

Universidade Estadual Paulista Julio de Mesquita Filho - UNESP  
Instituto de Biociências  
Departamento de Biodiversidade e Bioestatística  
Programa de Pós-Graduação em Biometria  
Métodos Numéricos e Computacionais

## **Trabalho Computacional 4 - Métodos Numéricos para Autovalores**

Aluno(a): Richard Castro Júnior

Professor: Daniela Renata Cantane

Botucatu-SP

2022

# 1 Introdução

Seja  $A$  uma matriz  $n \times n$ . Um escalar  $\lambda$  é chamado autovalor de  $A$  se existe um vetor diferente de zero  $\mathbf{x} \neq \mathbf{0}$ , chamado autovetor, tal que

$$A\mathbf{x} = \lambda\mathbf{x} \quad (1)$$

O autovetor  $\mathbf{x}$  deve ser diferente de zero, uma vez que  $\mathbf{x} = \mathbf{0}$  é uma solução trivial para a equação de autovalor (1) para todo escalar  $\lambda$  (OLVER; SHAKIBAN, 2018).

A equação de autovalor (1) é um sistema de equações lineares para as entradas do autovetor  $\mathbf{x}$  - desde que o autovalor  $\lambda$  seja especificado antecipadamente - mas é "ligeiramente" não linear como um sistema combinado para  $\lambda$  e  $\mathbf{x}$ . Reescrevendo a equação, temos na seguinte forma:

$$(A - \lambda I)\mathbf{x} = \mathbf{0} \quad (2)$$

sendo  $I$  é a matriz identidade e  $\lambda I\mathbf{x} = \lambda\mathbf{x}$  (OLVER; SHAKIBAN, 2018).

Para dado  $\lambda$ , a equação (2) é um sistema linear homogêneo para  $\mathbf{x}$ , e sempre tem a solução nula trivial  $\mathbf{x} = \mathbf{0}$ . Se um sistema linear homogêneo tem uma solução diferente de zero  $\mathbf{x} \neq \mathbf{0}$  se, e somente se, sua matriz de coeficientes  $(A - \lambda I \equiv p_A(\lambda))$ , é singular, e isto é necessário para que  $\lambda$  seja autovalor da matriz  $A_{n \times n}$ , tal que o posto  $< n$ , e os autovetores correspondentes são as soluções diferentes de zero para a equação de autovalor  $p_A(\lambda)\mathbf{x} = \mathbf{0}$  (ISAACSON; KELLER, 1994).

Assim, os autovalores podem, em princípio, ser encontrados como os zeros de  $p_A(\lambda)$  sem recorrer a nenhum dos autovetores. Dado algum autovalor,  $\lambda$ , então o autovetor correspondente é uma solução não trivial do sistema linear homogêneo (1). Por outro lado, se algum autovetor particular  $\mathbf{x}$  for conhecido, então o autovalor ao qual  $\mathbf{x}$  pode ser obtido tomando o produto interno de (1) com  $\mathbf{x}$  para obter (ISAACSON; KELLER, 1994)

$$\lambda = \frac{\mathbf{x}^* A \mathbf{x}}{\mathbf{x}^* \mathbf{x}} = \frac{1}{x_i} \sum_{j=1}^n a_{ij} x_j. \quad (3)$$

Uma decomposição de autovalores de uma matriz quadrada  $A$  é uma fatoração

$$A = X \Lambda X^{-1} \quad (4)$$

sendo  $X$  é não singular e  $\Lambda$  é diagonal (TREFETHEN; BAU III., 1997).

Esta definição pode ser reescrita

$$AX = X\Lambda \tag{5}$$

isto é,

$$\begin{bmatrix} & \\ & \\ A & \\ & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

Isso deixa claro que se  $x_m$  é a  $m$  a coluna de  $X$  e  $\lambda_m$  é a  $m$  a entrada diagonal de  $\Lambda$ , então  $Ax_m = \lambda_mx_m$ . Assim, a  $m$  a coluna de  $X$  é um autovetor de  $A$  e a  $m$  a entrada de  $\Lambda$  é o autovalor correspondente. Segundo Lloyd N. Trefethen, "a decomposição de autovalor expressa uma mudança de base para 'coordenadas de autovetor'"(TREFETHEN; BAU III., 1997).

Se  $Ax = b$  e  $A = X\Lambda X^{-1}$ , temos

$$X^{-1}b = \Lambda(X^{-1}x) \quad (6)$$

Assim, para calcular  $Ax$ , podemos expandir  $x$  na base das colunas de  $X$ , aplicar  $\Lambda$  e interpretar o resultado como um vetor de coeficientes de uma combinação linear das colunas de  $X$  (TREFETHEN; BAU III., 1997).

Supondo, então,  $A$  uma matriz arbitrária. Existe uma matriz não singular  $U$  com a propriedade de que

$$T = U^{-1}AU \quad (7)$$

sendo  $T$  é uma matriz triangular superior cujas entradas diagonais consistem nos autovalores de  $A$  (BURDEN; FAIRES, 2010). A matriz  $U$  satisfaz a condição  $\|U\mathbf{x}\|_2 = \|\mathbf{x}\|_2$  para qualquer vetor  $\mathbf{x}$ .

A matriz  $A_{n \times n}$  é simétrica se, e somente se, existir uma matriz diagonal  $D$  e uma matriz ortogonal  $Q$  com  $A = QDQ^t$ . Suponhamos aqui que  $A = QDQ^t$ , onde  $Q$  é ortogonal e  $D$  é diagonal (BURDEN; FAIRES, 2010). Então

$$A^t = (QDQ^t)^t = (Q^t)^t D Q^t = QDQ^t = A \quad (8)$$

e  $A$  é simétrica.

Toda matriz simétrica  $A$  pode ser escrita na forma  $A = QDQ^t$ , considerando os autovalores distintos de  $A$ . Se  $A\mathbf{v}_1 = \lambda_1\mathbf{v}_1$  e  $A\mathbf{v}_2 = \lambda_2\mathbf{v}_2$ , com  $\lambda_1 \neq \lambda_2$ , então desde  $A^t = A$  temos

$$(\lambda_1 - \lambda_2)\mathbf{v}_1^t\mathbf{v}_2 = (\lambda_1\mathbf{v}_1)^t\mathbf{v}_2 - \mathbf{v}_1^t(\lambda_2\mathbf{v}_2) = (A\mathbf{v}_1)^t\mathbf{v}_2 - \mathbf{v}_1^t(A\mathbf{v}_2) = \mathbf{v}_1^tA^t\mathbf{v}_2 - \mathbf{v}_1^tA\mathbf{v}_2 = 0, \quad (9)$$

então  $\mathbf{v}_1^t\mathbf{v}_2 = 0$  (BURDEN; FAIRES, 2010).

Portanto, podemos escolher vetores ortonormais para autovalores distintos simplesmente normalizando todos esses autovetores ortogonais. Quando os autovalores não são distintos, haverá subespaços de autovetores para cada um dos múltiplos autovalores, e com a ajuda do processo de ortogonalização de Gram-Schmidt podemos encontrar um conjunto completo de  $n$  autovetores ortonormais (BURDEN; FAIRES, 2010).

Agora, seja  $A$  seja uma matriz  $n \times n$  simétrica. Existem  $n$  autovetores de  $A$  que formam um conjunto ortonormal, e os autovalores de  $A$  são números reais. Se  $Q = (q_{ij})$  e  $D = (d_{ij})$ , então (BURDEN; FAIRES, 2010)

$$D = Q^tAQ = Q^{-1}AQ \quad (10)$$

implica que

$$AQ = QD \quad (11)$$

Seja  $1 \leq i \leq n$ ,  $\mathbf{v}_i = (q_{1i}, q_{2i}, \dots, q_{ni})^t$  a  $i$ -ésima coluna de  $Q$  e  $d_{ii}$  um autovalor de  $A$  com autovetor. Então

$$A\mathbf{v}_i = d_{ii}\mathbf{v}_i, \quad (12)$$

As colunas de  $Q$  são ortonormais, então os autovetores de  $A$  são ortonormais. Uma matriz simétrica cujos autovalores são todos números reais não negativos é algumas vezes chamada de definida não negativa (ou semidefinida positiva). Multiplicando esta equação à esquerda por  $\mathbf{v}_i^t$ :

$$\mathbf{v}_i^tA\mathbf{v}_i = d_{ii}\mathbf{v}_i^t\mathbf{v}_i \quad (13)$$

Desde  $\mathbf{v}_i^tA\mathbf{v}_i$  e  $\mathbf{v}_i^t\mathbf{v}_i$  sejam números reais e  $\mathbf{v}_i^t\mathbf{v}_i = 1$ , o autovalor  $d_{ii} = \mathbf{v}_i^tA\mathbf{v}_i$  é um número real, para cada  $i = 1, 2, \dots, n$  (BURDEN; FAIRES, 2010).

## 1.1 Método das Potências

O Método das Potências é uma técnica iterativa usada para encontrar aproximações de autovalores e determinar o autovalor dominante de uma matriz, isto é, o autovalor com a maior magnitude. Uma característica útil deste método é que ele produz não apenas um autovalor, mas também um autovetor associado (BURDEN; FAIRES, 2010).

Para aplicar o Método das Potências, assumimos que a matriz  $n \times n$   $A$  tem  $n$  autovalores  $\lambda_1, \lambda_2, \dots, \lambda_n$  com um associado coleção de autovetores linearmente independentes  $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \dots, \mathbf{v}^{(n)}\}$ . Além disso, assumimos que  $A$  tem precisamente um autovalor,  $\lambda_1$ , que é o maior em magnitude, de modo que

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0 \quad (14)$$

Se  $\mathbf{x}$  é qualquer vetor em  $\mathbb{R}^n$ , o fato de que  $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \dots, \mathbf{v}^{(n)}\}$  é linearmente independente implica que as constantes  $\beta_1, \beta_2, \dots, \beta_n$  existem com

$$\mathbf{x} = \sum_{j=1}^n \beta_j \mathbf{v}^{(j)} \quad (15)$$

Multiplicando ambos os lados desta equação por  $A, A^2, \dots, A^k, \dots$  dá

$$A\mathbf{x} = \sum_{j=1}^n \beta_j A\mathbf{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j \mathbf{v}^{(j)}, \quad A^2\mathbf{x} = \sum_{j=1}^n \beta_j \lambda_j A\mathbf{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j^2 \mathbf{v}^{(j)} \quad (16)$$

e geralmente,  $A^k\mathbf{x} = \sum_{j=1}^n \beta_j \lambda_j^k \mathbf{v}^{(j)}$ .

Se  $\lambda_1^k$  for fatorado de cada termo do lado direito da última equação, então

$$A^k\mathbf{x} = \lambda_1^k \sum_{j=1}^n \beta_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \mathbf{v}^{(j)} \quad (17)$$

Como  $|\lambda_1| > |\lambda_j|$ , para todo  $j = 2, 3, \dots, n$ , temos  $\lim_{k \rightarrow \infty} (\lambda_j/\lambda_1)^k = 0$ , e

$$\lim_{k \rightarrow \infty} A^k\mathbf{x} = \lim_{k \rightarrow \infty} \lambda_1^k \beta_1 \mathbf{v}^{(1)} \quad (18)$$

A sequência na equação (18) converge para 0 se  $|\lambda_1| < 1$  e diverge se  $|\lambda_1| > 1$ , desde que, é claro,  $\beta_1 \neq 0$ . Como consequência, as entradas no  $A^k\mathbf{x}$  crescerão com  $k$  se  $|\lambda_1| > 1$  e irão para 0 se  $|\lambda_1| < 1$ , talvez resultando em estouro ou estouro. Para cuidar dessa possibilidade, escalamos

as potências de  $A^k \mathbf{x}$  de maneira apropriada para garantir que o limite na equação (18) é finito e diferente de zero. A escala começa escolhendo  $\mathbf{x}$  para ser um vetor unitário  $\mathbf{x}^{(0)}$  relativo a  $\|\cdot\|_\infty$  e escolhendo um componente  $x_{p_0}^{(0)}$  de  $\mathbf{x}^{(0)}$  com

$$x_{p_0}^{(0)} = 1 = \|\mathbf{x}^{(0)}\|_\infty \quad (19)$$

O sucesso do método das potências está em assumir que  $A$  tem um único autovalor dominante de módulo máximo, que, por definição, é igual ao raio espectral:  $|\lambda_1| = \rho(A)$ .

O raio de convergência do método é governado pelo raio  $|\lambda_2/\lambda_1|$  entre os autovalores dominante e subdominante. Então, quanto mais longe o raio do autovalor dominante estiver do restante, mais rápido o método das potências irá convergir.

Uma vez que o resultado das iterações de  $A$  são, normalmente, muito grandes ( $\rho(A) > 1$ ) ou muito pequenos ( $\rho(A) < 1$ ) o método estará cada vez mais sujeito à overflow ou underflow e o método poderá parar antes de uma boa aproximação. Uma maneira de evitar isso é usar vetores unitários, em relação à norma euclidiana ou infinita, minimizando erros numéricos de computação.

Então, o algoritmo pode ser modificado como

$$\mathbf{u}^{(0)} = \frac{\mathbf{v}^{(0)}}{\|\mathbf{v}^{(0)}\|} \mathbf{e} \mathbf{u}^{(k+1)} = \frac{A \mathbf{u}^{(k)}}{\|A \mathbf{u}^{(k)}\|} \quad (20)$$

## 1.2 Decomposição QR

O primeiro passo é fatorar a matriz

$$A = A_0 = Q_0 R_0 \quad (21)$$

em um produto de uma matriz ortogonal  $Q_0$  e uma matriz positiva triangular superior (isto é, todas entradas da diagonal positivas)  $R_0$  usando o processo de ortogonalização de Gram-Schmidt.

O próximo passo é multiplicar essas matrizes trocando a ordem. O resultado é a nova matriz:

$$A_1 = R_0 Q_0 \quad (22)$$

Então repetimos esses dois passos usando o processo de Gram-Schmidt:

$$A_1 = Q_1 R_1 \quad (23)$$

Em seguida, multiplicamos na ordem inversa para obter

$$A_2 = R_1 Q_1 \quad (24)$$

O algoritmo pode ser escrito como:

$$A = A_0 = Q_0 R_0 \quad (25)$$

$$A_{k+1} = R_k Q_k = Q_{k+1} R_{k+1}, \quad k = 0, 1, 2, \dots, \quad (26)$$

sendo a matriz ortogonal subsequente  $Q_{k+1}$  e a matriz triangular superior  $R_{k+1}$  são calculadas diretamente de  $A_{k+1} = R_k Q_k$  aplicando o processo de Gram-Schmidt.

Para muitas matrizes  $A$  com todos autovalores reais, as iterações  $A_k$  convergem para uma matriz triangular superior  $V$  cujas entradas são os autovalores de  $A$ . Assim, após um número suficiente de iterações  $m$ , a matriz  $A_m$  terá elementos muito pequenos abaixo da diagonal, e tem-se um sistema completo de autovalores (aproximados) na sua diagonal.

Para cada autovalor, o cálculo do autovetor correspondente pode ser mais eficiente se realizado aplicando o método das potências inversas.

### 1.3 Tridiagonalização

Para aplicar o método QR, partimos de uma matriz simétrica em forma tridiagonal; ou seja, as únicas entradas diferentes de zero na matriz estão na diagonal ou nas subdiagonais diretamente acima ou abaixo da diagonal. Se esta não for a forma da matriz simétrica, o primeiro passo é aplicar o método de Householder para calcular uma matriz tridiagonal simétrica semelhante à matriz dada (BURDEN; FAIRES, 2010). Assim, primeiramente utilizamos o processo de tridiagonalização de Householder, e então aplicamos o método  $QR$  na matriz tridiagonal resultante.

Dessa forma, obtemos um algoritmo prático e eficiente para calcular autovalores de matrizes simétricas de grande porte, mas leva-se em consideração que este processo não preserva os autovalores da matriz, os elementos da diagonal de  $R$  não são os autovalores.

Aplicando o processo de Householder para a decomposição  $QR$ , podemos converter a matriz  $A$  na forma triangular superior  $R$  por uma sequência de matrizes elementares de reflexão.

Seja a matriz  $A_{n \times n}$  simétrica. Se  $H = I - 2\mathbf{u}\mathbf{u}^t$  é uma matriz de reflexão elementar, com  $\mathbf{u} \in \mathbb{R}^n$  um vetor unitário (na norma euclidiana), então  $A$  e  $B = HAH$  são matrizes similares e possuem

os mesmos autovalores - este é o lema do método de Householder. Note que  $H^{-1} = H$  e então  $B = H^{-1}AH$  é semelhante a  $A$ .

Usando a norma euclidiana, temos:

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{pmatrix}, \mathbf{y}_1 = \begin{pmatrix} 0 \\ +/- r_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ em que } r_1 = \|\mathbf{x}_1\| = \|\mathbf{y}_1\| \quad (27)$$

então  $\mathbf{x}_1$  contém todos os elementos fora da diagonal da primeira coluna de  $A$ .

Seja  $H_1 = I - 2\mathbf{u}_1\mathbf{u}_1^t$ , em que  $\mathbf{u}_1 = \frac{\mathbf{x}_1 - \mathbf{y}_1}{\|\mathbf{x}_1 - \mathbf{y}_1\|}$ , é a matriz de reflexão elementar correspondente de  $\mathbf{x}_1$  para  $\mathbf{y}_1$ . Uma boa escolha para o sinal de  $r_1$  é ser o oposto do sinal de  $a_{21}$ , que ajuda a minimizar os possíveis erros de arredondamento de  $\mathbf{u}_1$ .

Baseado no lema do método de Householder e no fato que o primeiro elemento de  $\mathbf{u}_1$  é zero, obtemos:

$$A_2 = H_1AH_1 = \begin{pmatrix} a_{11} & r_1 & 0 & \dots & 0 \\ r_1 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad (28)$$

para certos  $a_{ij}$ .

Então, por uma transformação de Householder, convertemos  $A$  em uma matriz semelhante  $A_2$  cuja primeira linha e coluna são da forma tridiagonal.

Repetimos o processo para a submatriz inferior direita  $(n-1) \times (n-1)$  de  $A_2$ . Temos que:

$$\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \\ a_{32} \\ a_{42} \\ \vdots \\ a_{n2} \end{pmatrix}, \mathbf{y}_2 = \begin{pmatrix} 0 \\ 0 \\ +/- r_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (29)$$



em que  $r_2 = \|\mathbf{x}_2\| = \|\mathbf{y}_2\|$ .

Seja  $H_2 = I - 2\mathbf{u}_2\mathbf{u}_2^t$ , em que  $\mathbf{u}_2 = \frac{\mathbf{x}_2 - \mathbf{y}_2}{\|\mathbf{x}_2 - \mathbf{y}_2\|}$ ,

Construímos a matriz semelhante:

$$A_3 = H_2 A_2 H_2 = \begin{pmatrix} a_{11} & r_1 & 0 & 0 & \dots & 0 \\ r_1 & a_{22} & r_2 & 0 & \dots & 0 \\ 0 & r_2 & \hat{a}_{33} & \hat{a}_{34} & \dots & \hat{a}_{3n} \\ 0 & 0 & a_{43} & a_{44} & \dots & a_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n3} & a_{n4} & \dots & a_{nn} \end{pmatrix} \quad (30)$$

em que as primeiras linhas e colunas são da forma tridiagonal agora. Os próximos passos do algoritmo seguem da mesma forma.

O resultado final é uma matriz  $T = A_n$  que tem os mesmos autovalores, mas não os mesmos autovetores, da matriz simétrica original  $A$ .

## 1.4 Decomposição SVD

A decomposição em valor singular (SVD) é a generalização da fatorização espectral para matrizes não simétricas, usada para definir a pseudoinversa de uma matriz, na qual promove mecanismo para "inverter" matrizes singulares e não quadradas, e uma alternativa para construção de solução de mínimos quadrados para sistemas lineares em geral.

Primeiramente, definimos valores singulares  $\sigma_1, \dots, \sigma_r$  de uma matriz  $A_{m \times n}$  são raízes quadradas positivas  $\sigma_i = \sqrt{\lambda_i} > 0$ , de autovalores não nulos da matriz  $A^t A$ , em que  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ . Os correspondentes autovetores de  $A^t A$  são conhecidos como vetores singulares de  $A$ .

Uma matriz  $A_{m \times n}$  real não nula com posto  $r > 0$  pode ser fatorada da forma

$$A = P \Sigma Q^t \quad (31)$$

um produto de uma matriz  $P_{m \times r}$  com colunas ortonormais, então  $P^t P = I$ , e  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$  matriz diagonal  $r \times r$  que tem como elementos da sua diagonal valores singulares de  $A$ , e uma matriz  $Q^t_{r \times n}$  com linhas ortonormais, então  $Q^t Q = 1$ .

Temos que  $AQ = P\Sigma$ , então

$$A\mathbf{q}_i = \sigma_i \mathbf{p}_i, \quad i = 1, \dots, r, \quad (32)$$

ou seja, o objetivo é encontrar vetores  $\mathbf{p}_i$  e  $\mathbf{q}_i$ , com  $i = 1, \dots, r$ , que satisfazem (32).

Seja  $\mathbf{q}_i$   $i = 1, \dots, r$  autovetores ortonormais de  $A^t A$  correspondentes aos autovalores não nulos, que formam uma base de dimensão  $r = \text{pos } A^t A = \text{pos } A$ . Então, pela definição de valores singulares

$$A^t A \mathbf{q}_i = \sigma_i^2 \mathbf{q}_i, \quad i = 1, \dots, r \quad (33)$$

Os vetores  $\mathbf{w}_i = A\mathbf{q}_i$  são ortogonais:

$$\mathbf{w}_i \cdot \mathbf{w}_j = \mathbf{w}_i^T \mathbf{w}_j = (A\mathbf{q}_i)^T A\mathbf{q}_j = \mathbf{q}_i^T A^T A \mathbf{q}_j = \sigma_j^2 \mathbf{q}_i^T \mathbf{q}_j = \sigma_j^2 \mathbf{q}_i \cdot \mathbf{q}_j = \begin{cases} 0, & i \neq j, \\ \sigma_i^2, & i = j \end{cases} \quad (34)$$

Temos também que  $\|\mathbf{w}_i\| = \sqrt{w_i w_i} = \sigma_i$ ,  $i = 1, \dots, r$ . Então, podemos concluir que

$$\mathbf{p}_i = \frac{\mathbf{w}_i}{\sigma_i} = \frac{A\mathbf{q}_i}{\sigma_i}, \quad i = 1, \dots, r \quad (35)$$

formam um conjunto de vetores ortonormais que satisfazem (32).

Se  $A$  tem valores singulares distintos, sua decomposição em valor singular (31) é quase única, trocando os sinais simultaneamente de uma ou mais colunas de  $Q$  e  $P$  correspondentes. Usando a transposta de (3) e notando que  $\Sigma^t = \Sigma$ , pois é uma matriz diagonal, obtemos

$$A^t = Q \Sigma P, \quad (36)$$

que é uma decomposição em valor singular da matriz  $A^t$ .

Em particular, temos que uma matriz  $A$  e sua transposta  $A^t$  tem os mesmos valores singulares, onde os autovetores não são os mesmos (os de  $A$  são colunas ortonormais de  $Q$ , enquanto de  $A^t$  são colunas ortonormais de  $P$ ):

$$A^t \mathbf{p}_i = \sigma_i \mathbf{q}_i, \quad i = 1, \dots, r \quad (37)$$

Além disso, a SVD serve para diagonalizar a matriz  $A^t A$ . Desde que  $P^t P = I$ ,

$$Q^t A^t A Q = Q^t A^t P^t P A Q = (P A Q)^t (P A Q) = \Sigma^t \Sigma = \Sigma^2. \quad (38)$$

Se  $A$  tem posto máximo igual a  $n$ , então  $Q$  é uma matriz ortogonal  $n \times n$  e então (38) implica que a transformação linear da matriz  $A^t A$  é diagonalizada quando expressa em termos da base ortonormal formada por vetores singulares.

Se  $r = \text{pos}(A) < n$ , então podemos completar os  $r$  vetores singulares  $\mathbf{u}_1, \dots, \mathbf{u}_r$  com  $n - r$  vetores unitários  $\mathbf{u}_{r+1}, \dots, \mathbf{u}_n \in \text{nul}(A)$  para formar uma base ortonormal do  $\mathbb{R}^n$

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & \dots & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \sigma_r & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad (39)$$

em que  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$  e  $r \leq \min\{m, n\}$ .

Se  $A$  é uma matriz quadrada não singular, a pseudoinversa é sua própria inversa:

$$A^{-1} = (P\Sigma Q^t)^{-1} = (Q^{-1})^t \Sigma^{-1} P^{-1} = Q\Sigma^{-1} P^t = A^+. \quad (40)$$

então, a pseudoinversa de uma matriz  $m \times n$  com decomposição SVD  $A = P\Sigma Q^t$  é uma matriz  $n \times m$   $A^+ = Q\Sigma^{-1} P^t$  - lembrando que a inversa de uma matriz ortogonal é igual a sua transposta.

Por isso, seja  $A$  uma matriz  $m \times n$  matriz de posto  $n$ . Então,

$$A^+ = (A^t A)^{-1} A^t. \quad (41)$$

## 2 Objetivos

Implementar computacionalmente os métodos numéricos para autovalores: Método das Potências, Decomposição QR, Triadiagonalização de Householder e Decomposição em Valor Singular (SVD) e comparar os resultados em relação à quantidade de iterações, ao tempo computacional e ao erro final do cálculo. Implementar a Decomposição SVD para melhoramento de imagem.

### 3 Procedimento

Os métodos numéricos para autovalores (Método das Potências, Decomposição QR, Triadiagonalização de Householder e SVD) foram implementados utilizando o MATLAB ®. Trata-se de uma linguagem de programação que expressa diretamente matrizes e arranjos matemáticos, combinando um ambiente de desktop ajustado para análise iterativa e processos de design com uma linguagem de programação.

Foi utilizada matrizes extraídas do site <https://math.nist.gov/MatrixMarket/>, que fornece acesso a um repositório de dados de teste para uso em estudos comparativos de algoritmos para álgebra linear numérica. Deste modo, foram escolhidas as matrizes *blackhole* (2132 x 2132), *bcsstk22* (138 x 138) e *bcsstk02* (66 x 66). Estas foram analisadas a fim de estabelecer uma comparação entre os dados obtidos, observando o comportamento desses métodos numéricos para autovalores.

Foi selecionada uma imagem da obra Cristo no Templo (1897), de Heinrich Hofmann para fazer o melhoramento dela atrás da Decomposição SVD. A imagem tem dimensões 192x262 em unidade pixels. O melhoramento começou com um reordenamento das dimensões da imagem e, em seguida, é escolhido o nível de qualidade da imagem. Este procedimento foi repetido até fosse obtido um número satisfatório de resultados para comparação de imagens e análise dos valores singulares da imagem.

#### 3.1 Matriz de Hilbert

Uma função que receba um inteiro  $n$  e que retorne uma matriz de Hilbert de ordem  $n$  foi implementada e testados para todos os programas implementados utilizando a matriz de Hilbert para resolver o sistema linear com

$$a_{ij} = \frac{1}{i+j-1} \quad b = \frac{1}{3} \sum_{j=1}^n a_{ij}$$

Foram considerados  $n = 5$ ,  $n = 7$ ,  $n = 9$ ,  $n = 10$  e  $n = 12$ , cujos resultados foram analisados e comparados. Este valores foram escolhidos baseados no Trabalho Computacional 1 - Métodos Diretos para Sistemas Lineares, onde foi verificado que, a partir da ordem 29, a matriz de Hilbert é singular, isto é, não possuem matriz inversa e, pela Decomposição de Cholesky, a partir da ordem 13, a matriz não é definida positiva.

Implementados os cálculos para as matrizes de Hilbert, os resultados foram organizados de acordo com o método implementados. Foram comparados os resultados para o código do *script* e a

função `hilb(n)` do MATLAB ® e foram analisados o tempo computacional, o número de iterações da resolução do sistema, ou seja, quantas iterações foram necessárias para obter a solução aproximada, e a precisão da resolução, a partir do erro relativo da última iteração ( $\epsilon$ ).

## 4 Resultados e discussão

### 4.1 Matriz de Hilbert

Nas tabelas 1, 2, 3 e 4 estão os resultados para os cálculos efetuados com a matriz de Hilbert. Conforme a tabela 1, vemos que a comparação feita não é entre o algoritmo elaborado para a matriz de Hilbert e o código do MATLAB ®, mas entre os cálculos pelo métodos de potências que começam pelo vetor estabelecido, que começa com um elemento 1 e todos os outros zero, de acordo com a ordem da matriz, e os cálculos começam com uma matriz aleatório, produzida pelo próprio algoritmo através de um código do programa utilizado.

Tabela 1: Implementação do método de potências.

	Vetor (1 0 <sup>n</sup> )			Vetor inicial aleatório		
n	Tempo computacional (10 <sup>-4</sup> s)	Número de iterações	Erro relativo final	Tempo computacional (10 <sup>-4</sup> s)	Número de iterações	Erro relativo final
5	2.5830	19	1.4687e-16	0.5570	17	4.7429e-16
7	18.6070	22	1.4153e-16	0.6670	21	1.241e-16
9	3.6340	23	9.2055e-17	0.7340	22	7.3434e-17
10	0.7160	24	1.5202e-16	0.6320	23	6.799e-17
12	1.044	25	1.127e-16	0.6230	24	1.5454e-16

Foi escolhido esta análise para estes resultados, pois o algoritmo elaborado e o código `hilb(n)` não demonstraram resultados com diferenças significativas. Em contrapartida, observamos diferenças mais evidentes entre o tempo computacional entre os resultados para os respectivos algoritmos.

Quando o cálculo iniciou com o um vetor aleatório, o tempo computacional não chegou a uma unidade de tempo, na escala de 10<sup>-4</sup>. A maior diferença dentre estes tempos foi para a matriz de ordem 7 (96.42%). Na maioria dos cálculos, a melhor precisão ocorreu quando o cálculo iniciava com o vetor preestabelecido, mas essa diferença não tão significativa quanto o tempo computacional. O mesmo pode ser dito para o número de iterações, sendo que todos tem uma diferença de 1 iteração, com exceção do cálculo para a matriz de ordem 5 (diferença de 2 iterações). Observando a tabela 1, os melhores resultados foram para as matrizes de ordem 9 e 10 com cálculos iniciados a partir de vetores aleatórios, considerando os tempos computacionais relativamente baixos, a pouca diferença

de número de iterações e as maiores precisões (de acordo com os menores erros) apresentados.

As outras tabelas apresentam os resultados de cálculos que utilizaram o algoritmo da matriz de Hilbert e o código do MATLAB ®.

Tabela 2: Implementação do método de decomposição QR.

n	Código implementado			Função <i>hilb</i> ( <i>n</i> )		
	Tempo computacional (10 <sup>-3</sup> s)	Número de iterações	Erro relativo final	Tempo computacional (10 <sup>-3</sup> s)	Número de iterações	Erro relativo final
5	7.8153	6	3.7270e-16	10.7881	6	3.7270e-16
7	6.5395	6	0.0000	25.5789	6	0.0000
9	14.0362	7	2.1344e-16	8.8457	7	2.1344e-16
10	4.6297	7	2.0817e-16	7.8194	7	2.0817e-16
12	4.4488	8	1.9965e-16	12.9717	8	1.9965e-16

Como se vê na tabela 2, não há diferença de número de iterações, nem de precisão do algoritmos entre os cálculos feitos com o código implementado para a matriz e o a função do programa. Isto é averiguado também nas tabelas 3 e 4 e, deste modo, a comparação é feita a partir do tempo computacional. Não obstante, ainda é possível observa o comportamento de cada métodos para o cálculo dos autovetores das matrizes de Hilbert.

Duas coisas chamam mais a atenção na tabela 2: a precisão do cálculo para a matriz de ordem 7 e o tempo computacional dos cálculos cujas matrizes foram elaboradas a partir do algoritmo elaborado. Pode-se dizer que o cálculo dos autovalores para a matriz de ordem 7 apresentou valores tão próximos que o próprio programa do cálculo apresentou erro igual zero ou que, de fato, há uma exatidão dos autovalores encontrados. Além disso, foi para este cálculo que, em comparação dos resultados, houve a maior diferença entre os tempos computacionais (74.43%).

Os cálculos pela decomposição QR mantiveram de certa forma a precisão para os valores dos autovalores e mostraram uma redução do número de iterações, sendo, assim, mais econômico computacionalmente dos que o método de potências. Entretanto, os cálculos pelo método de potências são mais rápidos, ainda mais considerando cálculos iniciados com vetores aleatórios.

Já na tabela 3, são apresentados os resultados para os cálculos feitos a partir da tridiagonalização de Householder. Primeiramente, apontamos que, a partir de  $n = 10$ , todas as matrizes de ordem par têm uma matriz tridiagonal que não definida-positiva. Para esses casos, não foi possível calcular os autovalores, pois, aplicando o processo de Householder para a decomposição QR, o algoritmo QR preserva a classe de matrizes simétricas tridiagonais e, dada uma matriz  $A$  simétrica, a forma quadrática é definida como  $Q_A(x) = x^T A x$ , para  $x \in \mathbb{R}^n$ . Se  $\forall x \neq 0, Q_A(x) > 0$ , então a

forma quadrática é chamada definida-positiva (BHAYA, 2022).

Tabela 3: Implementação do método de triadiagonalização de Householder.

n	Código implementado			Função $h_{ilb}(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
5	27.9609	6	6.1655e-01	13.9050	6	6.1655e-01
7	3.2809	5	1.7672e-01	13.5847	5	1.7672e-01
9	11.5429	7	1.4805e-01	2.7638	7	1.4805e-01
10	Matriz não definida positiva					
11	13.8993	9	1.3138e-01	17.6326	9	1.3138e-01
12	Matriz não definida positiva					
13	6.4941	11	1.2033e-01	6.8671	11	1.2033e-01

Em comparação aos dois anteriores métodos, não demonstrou ser preciso para as matrizes de Hilbert, além de ter um tempo consideravelmente maiores, em relação aos outros métodos. Por este método também não há diferenças de precisão e número de iterações em os cálculos feitos com o algoritmo elaborado para a matriz e com a função  $h_{ilb}(n)$ . Podemos observar também que, para ordens maiores de matriz, o tempo computacional é menor, assim como o erro do cálculo, ou seja, há maiores precisões e eficiência para matrizes maiores. Em contrapartida, há um maior gasto computacional no que consta o crescente número de iterações conforme o aumento da ordem das matrizes.

Os cálculos em que foi aplicado o método SVD, apresentaram os resultados mais consistentes, conforme se vê na tabela abaixo:

Tabela 4: Implementação do método SVD.

n	Código implementado			Função $h_{ilb}(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
5	2.0946	1	3.0652e-16	4.5797	1	3.0652e-16
7	62.1224	1	3.3614e-16	9.7704	1	3.3614e-16
9	1.2893	1	2.3071e-16	66.8300	1	2.3071e-16
10	6.8704	1	3.5047e-16	72.1000	1	3.5047e-16
12	0.0932	1	4.8343e-16	0.08975	1	4.8343e-16

Por este método, todos os cálculos foram finalizados com apenas uma iteração. É, deste modo, um método que viabiliza o custo computacional dos cálculos de autovalores. Não necessariamente são cálculos mais rápidos, e é interessante notar que, como os resultados dos cálculos que utilizaram a triadiagonalização QR, com a matriz de ordem 12, obtivemos os menores tempos computacionais (escala de  $10^{-5}$ ). As precisões dos resultados nesses cálculos não apresentaram mudanças signifi-

cativas e, disso, vemos que a aplicação do método SVD forneceu resultados mais sólidos.

## 4.2 Modelos matemáticos com autovalores

As tabelas 5, 6, 7 e 8 mostram os resultados dos cálculos de autovalores, onde foram utilizadas matrizes maiores que as matrizes de Hilbert anteriormente, sendo uma delas densa (*bcsstk02*) e outras dois esparsas (*blkhole* e *bcsstk22*). Os cálculos foram efetuados igualmente aos precedentes.

Tabela 5: Implementação do método de potências.

Matriz	Vetor (1 0 <sup>n</sup> )			Vetor inicial aleatório		
	Tempo computacional (10 <sup>-3</sup> s)	Número de iterações	Erro relativo final	Tempo computacional (10 <sup>-3</sup> s)	Número de iterações	Erro relativo final
blkhole	40.0140	692	6.1089e-16	81.0530	1521	2.9784e-16
bcsstk22	6.6289	991	5.6855e-16	12.2942	917	6.1255e-16
bcsstk02	4.5028	289	5.3311e-13	8.5138	299	2.3207e-13

O comportamento do método de potências é diferente neste caso, pois, de acordo com a tabela 5, apenas no cálculo com a matriz *bcsstk22* o número de iterações foi maior quando utilizou inicialmente o vetor preestabelecido. Por esses resultados, os cálculos iniciados com vetores aleatórios apresentaram maior precisão, no entanto as diferenças de erro são menores e, deste modo, menos significativas em comparação as matrizes de Hilbert, cujas ordem eram menores. Além disso, os cálculos com o vetor preestabelecido foram mais rápidos, diferentemente dos cálculos da matriz de Hilbert, e, para ambos os casos, quanto maior a ordem da matriz, maior foi o tempo computacional.

Para os cálculos efetuado com a decomposição QR, não foi possível calcular os autovalores para a matriz *blkhole*. Outras matrizes foram testadas também, mas com nenhuma do banco de dados usado foi possível operar. Entretanto, foi possível observar o comportamento desse método a partir das outras matrizes.

Tabela 6: Implementação do método de decomposição QR.

Matriz	Tempo computacional (s)	Número de iterações	Erro relativo final
blkhole	ERROR		
bcsstk22	1.7860	57	4.5280e-16
bcsstk02	0.3549	34	5.7242e-16

Os resultados mostram a mesma tendência que os cálculos com método de potências: conforme a ordem da matriz aumenta, o tempo computacional também é maior. Este comportamento não é observado com as matrizes de Hilbert, pois os cálculos dos autovalores com essas matrizes



demonstraram mais o comportamento referente às próprias matrizes. Tal fato pode estar associado ao malcondicionamento das matrizes, devido a quase singularidade dos coeficientes.

Do mesmo modo, o número de iterações foi maior para o cálculo com a matriz de maior ordem, sendo esta, então, a que tem maior gasto computacional, mas também foi o cálculo mais preciso, embora não seja uma diferença tão significativa comparando com a matriz de menor ordem.

Já os resultados da tabela 7 mostram resultados que permitem uma associação interessante aos resultados dos cálculos com a matriz de Hilbert. Vemos que, quanto maior a ordem da matriz, menores são os erros dos cálculos. A matriz utilizadas nestes cálculos são expressivamente maiores que as matrizes de Hilbert utilizadas, logo é bem justificável as diferenças notadamente menores, e disso podemos estabelecer que, embora a precisão do método de triadiagonalização de Householder não seja boa, quanto maiores as matrizes utilizadas, maiores serão as precisões. Além disso, o tempo de cálculo foi bem menor em comparação ao método de decomposição QR, assim como menores números de iteração comparados ao método de potências.

Tabela 7: Implementação do método de triadiagonalização de Householder.

Matriz	Tempo computacional (s)	Número de iterações	Erro relativo final
blkhole	271.4401	2130	1.2466e-03
bcsstk22	0.7475	136	6.4880e-03
bcsstk02	0.1175	64	2.8219e-02

Tabela 8: Implementação do método SVD.

Matriz	Tempo computacional (s)	Número de iterações	Erro relativo final
blkhole	23.0344	1	4.4494e-15
bcsstk22	0.0377	1	1.8033e-15
bcsstk02	0.0262	1	1.1312e-15

Não obstante, o melhores resultados forma, de fato, apresentados pelos cálculos onde foi implementado o método SVD. Assim como para a matriz de Hilbert e conforme a tabela 8, neste caso , foi necessário apenas uma iteração para obter o resultado dos autovalores. Embora não tenham tido os menores erros, ainda são resultado expressivamente bem precisos e muito rápidos, quando comparados aos métodos de triadiagonalização de decomposição QR, estando na faixa de tempo do método de potências. Os resultados também mostram que, quanto maior a ordem da matriz, maior é o tempo computacional e menor será a precisão, o que potencializa o método usado em cálculos com matrizes de menores ordens.

### 4.3 Correção de imagem

A decomposição em valor singular (SVD) é uma excelente alternativa para construir soluções de mínimos quadrados para sistemas lineares em geral, uma vez que promove mecanismo, que define uma matriz pseudoinversa (a partir da matriz  $A^t A$ ), para fazer cálculos com matrizes singulares e não quadradas. Esta decomposição é uma ferramenta interessante para redimensionar uma imagem e controlar sua qualidade, justamente pensando em quaisquer imagens, que não são necessariamente em dimensões de altura e largura iguais (ou pixels que formam uma matriz quadrada). Analisamos as figuras 1-8, onde temos as reconstruções da imagem selecionada para este estudo.

Conforme as dimensões, em pixels, da imagem, a primeira reconstrução foi feita de acordo com as dimensões da obra original. A figuras 1 mostra a imagem reconstruída e o gráficos dos valores singulares (elementos diagonais da matriz diagonal) da imagem.

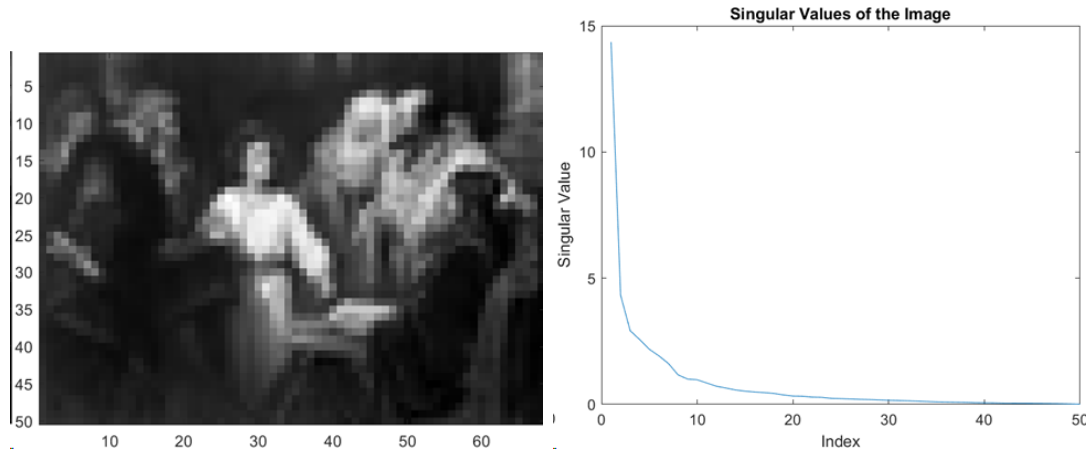


Figura 1: Imagem 58x68 com k=50 valores singulares e valores singulares da imagem.

O maior número de valores singulares possíveis para cada reconstrução é igual ao módulo da menor dimensão da imagem. Para a figura, vemos que dimensões pequenas não permitem que tenhamos uma boa definição da imagem. Deste modo, a análise da reconstrução seria mais interessante visando uma imagem com maiores dimensões.

Mantendo as mesmas proporções das dimensões, foi escolhida a reconstrução da imagem pelas dimensões 480x512, conforme as imagens 2-7.

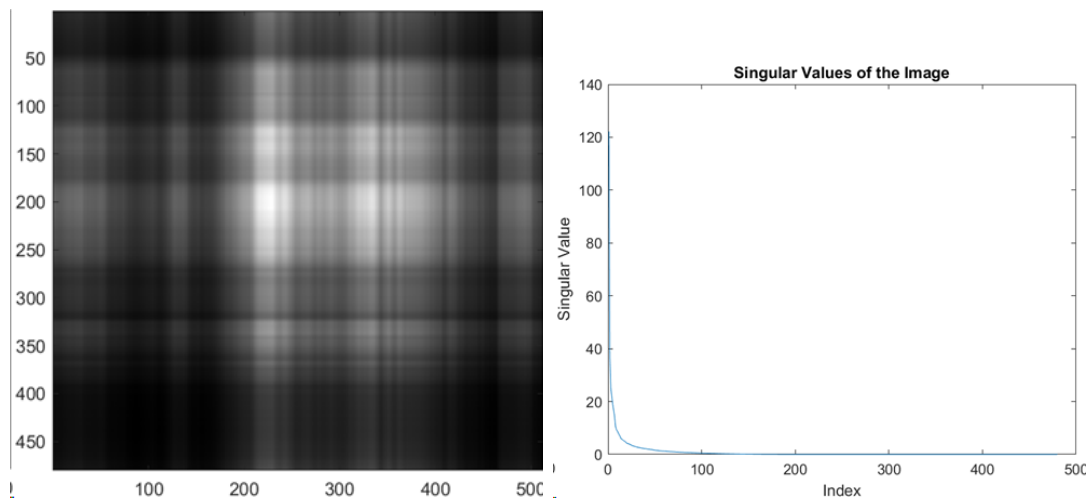


Figura 2: Imagem 480x512 com  $k=1$  valor singular e valores singulares da imagem.

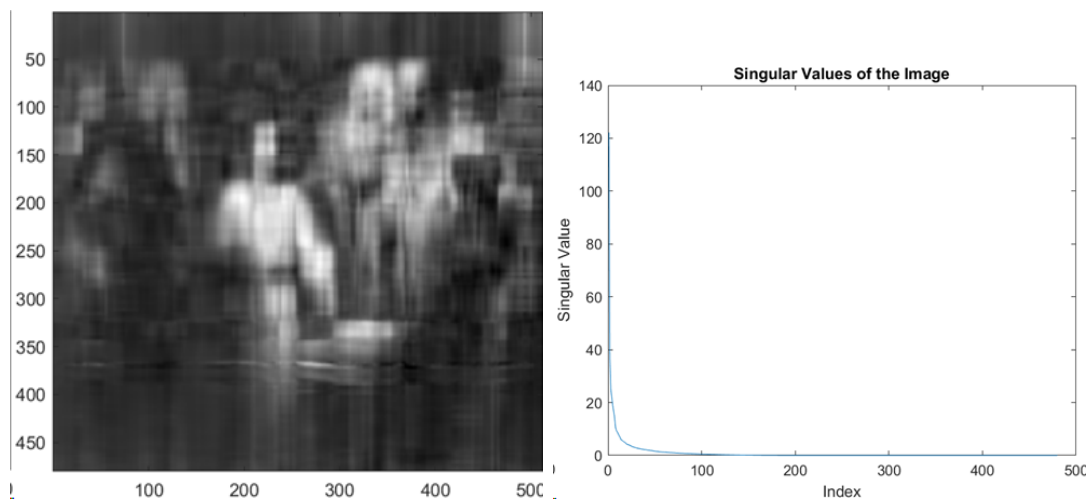


Figura 3: Imagem 480x512 com  $k=10$  valores singulares e valores singulares da imagem.

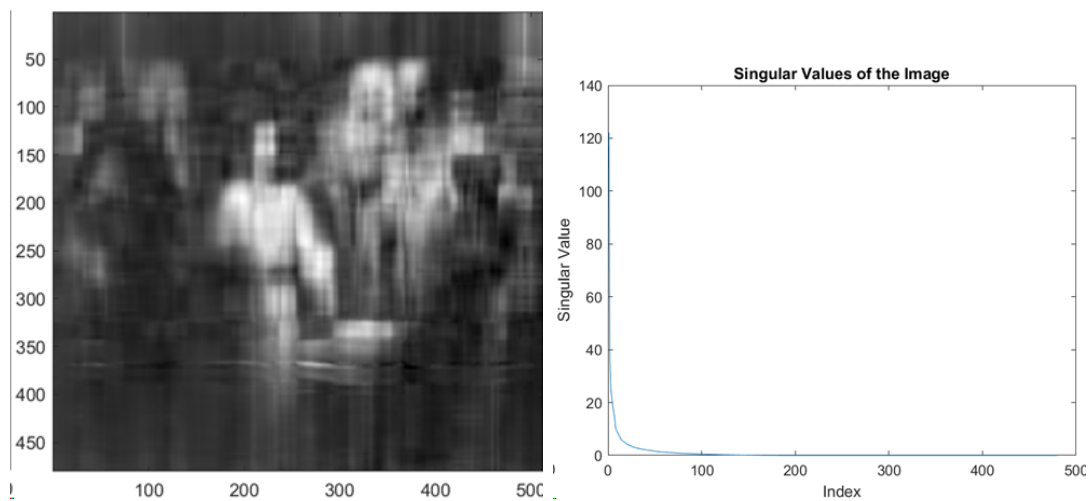


Figura 4: Imagem 480x512 com  $k=100$  valores singulares e valores singulares da imagem.

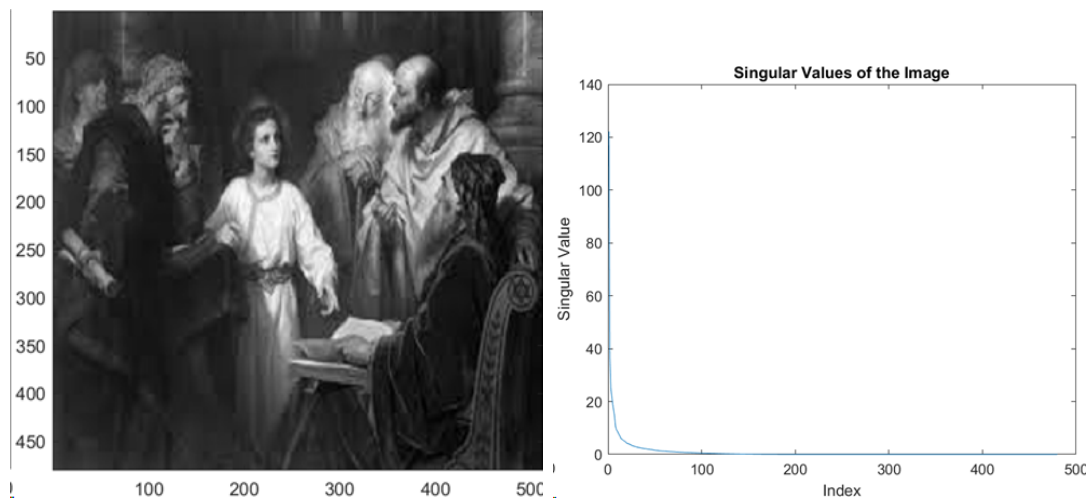


Figura 5: Imagem 480x512 com k=250 valores singulares e valores singulares da imagem.

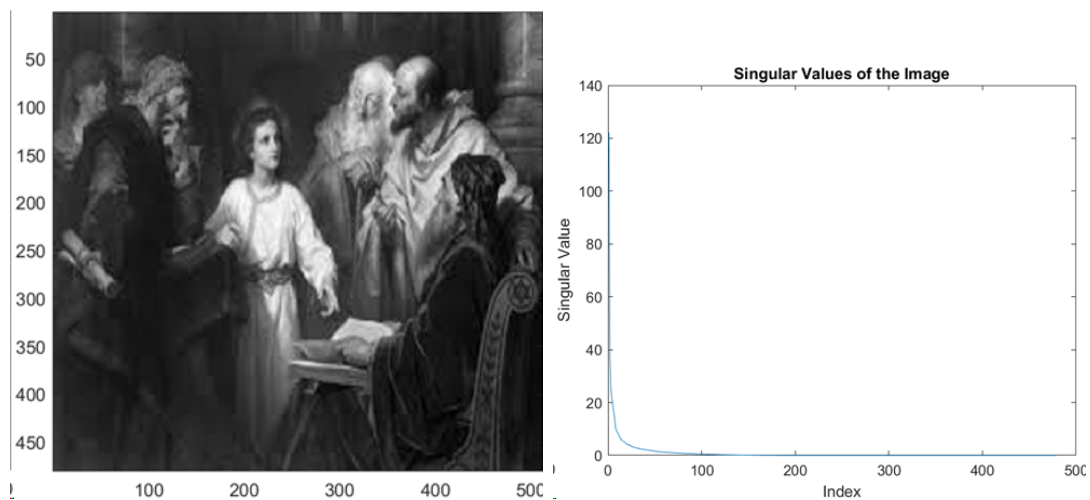


Figura 6: Imagem 480x512 com k=400 valores singulares e valores singulares da imagem.

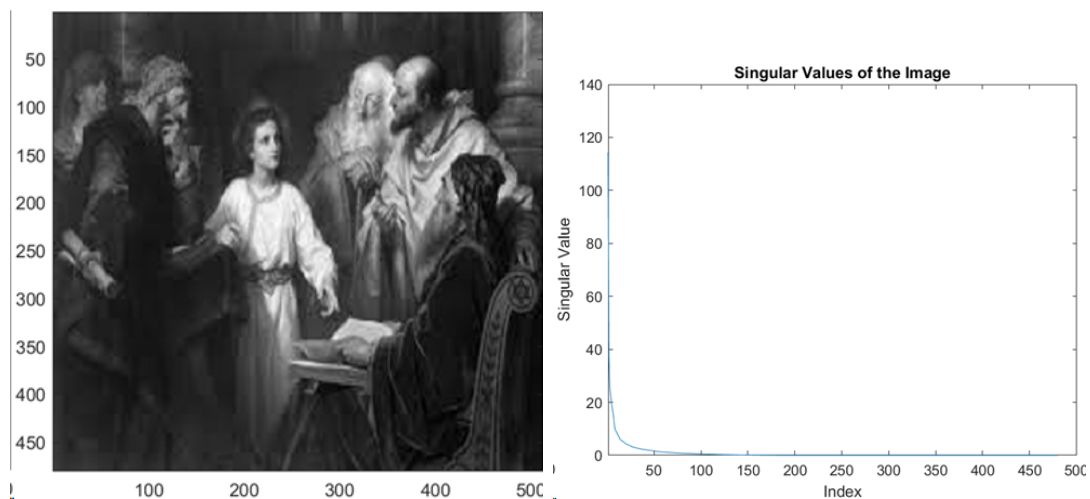


Figura 7: Imagem 480x512 com k=480 valores singulares e valores singulares da imagem.

Escolhendo valores singulares de 1 a 480, nitidamente, a imagem vai recebendo melhoras em sua qualidade e definição. Assim é possível distinguir mais detalhes da figura e ter mais clareza. Além disso, a figura 8 é uma redefinição de maiores dimensões da imagem, mantendo as mesmas proporções, onde temos uma imagem mais clara e com detalhes mais evidências que a figura 7.

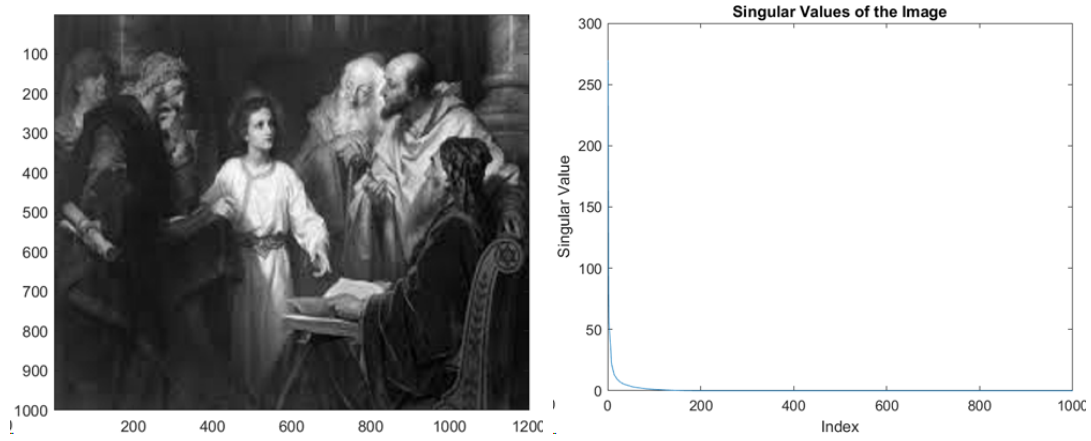


Figura 8: Imagem 1000x1200 com k=1000 valores singulares e valores singulares da imagem.

Seguindo as equações 31-41, a partir do teorema Eckart-Young (ECKART; YOUNG, 1936), a matriz  $A^t A$  é obtida pela fatoração SVD da matriz A, ou seja:

$$A^t A = Q \Sigma P^T = \sum_{i=1}^k \sigma_i q_i p_i. \quad (42)$$

Disso, a tabela 9 mostram os números reais armazenados em cada imagens, onde, aquelas que tem maiores números reais armazenados, são as imagens de melhor qualidade e definição:

Tabela 9: Números reais armazenados nas imagens.

Dimensão da imagem	k=1	k=10	k=50	k=100	k=250	k=400	k=480	k=1000
50x68	-	-	5900	-	-	-	-	-
480x512	992	9920	-	99200	248000	396800	476160	-
1000x1200	-	-	-	-	-	-	-	2200000

## 5 Conclusão

Os métodos numéricos para autovalores foram implementados computacionalmente e foram feitas comparações entre todos os casos em relação ao número de iterações, tempo computacional e erro relativo final para a matriz de Hilbert e para três matrizes escolhidas. Os cálculos que utilizaram os métodos de potências mostraram que cada caso apresenta uma eficiência maior em relação à precisão dos cálculos, ao passo que, para o método de tridiagonalização de Householder, apesar de baixa precisão, apresentou os resultados mais uniforme em razão do melhor custo computacional. Mas, de fato, o melhor método utilizado foi a decomposição de valores singulares, uma vez que só foi necessário uma iteração para todas as operações e este método permite cálculos com matrizes não singulares e não quadradas, com tempos baixos de cálculos, pequeno número de iterações e boa precisão. Inclusive, o método SVD permite o redimensionamento de imagens e controle de qualidade a partir dos valores singulares de uma matriz diagonal, tal que, quanto maiores as dimensões de uma imagem e o número de valores singulares, melhores será a clareza e definição da mesma.

## 6 Referências

- BHAYA, Amit. Matrizes positivas definidas, semidefinidas, etc. Disponível em: <http://www.nacad.ufrj.br/~amit/>. Acesso em: 16 fev. 2022.
- BURDEN, Richard L.; FAIRES, J. Douglas. Numerical Analysis. 9. ed. Boston: Cengage Learning, 2010.
- ECKART, C.; YOUNG, G. The Approximation of One Matrix by Another of Lower Rank, Psychometrika. v. 1, n. 3, p. 211– 218, 1936.
- ISAACSON, E.; KELLER, H. B. Analysis of Numerical Methods. Dover: New York, 1994.
- OLVER, Peter J.; SHAKIBAN, Chehrzad. Applied Linear Algebra. 2. ed. San Francisco: Springer, 2018.

TREFETHEN, Lloyd N.; BAU III., David Bau. Numerical linear algebra. Philadelphia: SIAM, 1997.