

# Cálculo de autovalores e aplicações.

Exercício Computacional  
MAP3122 - Quadrimestral 2022  
Prof. Antoine Laurain

Esse exercício computacional deve ser feito em duplas. Veja as instruções detalhadas no final do texto.

## 1 Introdução

Problemas de autovalor de matriz surgem em um grande número de disciplinas na ciência e na engenharia. Eles constituem a ferramenta básica usada na concepção de edifícios, pontes e turbinas que são resistentes a vibrações. Eles permitem modelar redes de filas e analisar a estabilidade de redes elétricas ou fluxo de fluidos. Eles também permitem ao cientista entender fenômenos físicos locais ou estudar os padrões de bifurcação em sistemas dinâmicos.

Os autovalores de uma matriz  $A$  podem ser determinados encontrando as raízes do polinômio característico da matriz. Isso é fácil para matrizes  $2 \times 2$ , mas a dificuldade aumenta rapidamente com o tamanho da matriz. Por esse motivo, é importante desenvolver algoritmos numéricos eficientes para calcular aproximações numéricas de autovalores. Métodos eficientes e precisos para calcular autovalores e autovetores de matrizes arbitrárias não eram conhecidos até que o algoritmo QR foi inventado em 1961. Em muitas aplicações,  $A$  pode ser simétrica, tridiagonal, ou ter alguma forma específica. Por isso, muitos métodos numéricos foram criados para tipos de matrizes específicos.

Neste exercício, estudaremos e implementaremos três algoritmos diferentes para calcular autovalores e autovetores: o método das potências, o método da potência inverso e o algoritmo QR. Depois veremos uma aplicação ao estudo de propriedades de grafos.

### 1.1 Algumas notações

Dado um vetor  $u \in \mathbb{R}^n$ , usamos a convenção de que ele é um vetor coluna. Isto significa que  $u$  é identificado a uma matriz de  $\mathbb{R}^{n \times 1}$ , isto é, uma matriz com  $n$  linhas e uma coluna. Assim, o transposto  $u^T$  é uma matriz de  $\mathbb{R}^{1 \times n}$ , isto é, uma matriz com  $n$  colunas e uma linha. Dados dois vetores  $u, v$  de  $\mathbb{R}^n$ , o produto  $uv^T$  segue a regra de cálculo matricial de uma matriz  $\mathbb{R}^{n \times 1}$  com uma matriz de  $\mathbb{R}^{1 \times n}$ , então  $uv^T \in \mathbb{R}^{n \times n}$  é uma matriz quadrada. O produto  $uv^T$  também é chamado *produto externo* ou *produto tensorial* de  $u$  e  $v$  e às vezes é notado  $u \otimes v$ .

O produto escalar (ou produto *interno*) de dois vetores  $u, v$  de  $\mathbb{R}^n$  é denotado por  $u \cdot v$  e definido como

$$u \cdot v = u^T v = \sum_{i=1}^n u_i v_i.$$

A norma  $\|u\|$  de um vetor  $u \in \mathbb{R}^n$  é a norma euclidiana, definida como

$$\|u\| = \sqrt{u \cdot u} = \sqrt{\sum_{i=1}^n u_i^2}.$$

### 1.2 Autovalores e autovetores

Dada uma matriz quadrada  $A \in \mathbb{R}^{n \times n}$ , dizemos que  $\lambda \in \mathbb{C}$  é um *autovalor* de  $A$  se existir um vetor  $x \in \mathbb{C}^n$ ,  $x \neq 0$ , tal que  $Ax = \lambda x$ . Nesse caso chamamos  $x$  de *autovetor* associado ao autovalor  $\lambda$ . Observe que apesar de  $A$  ter suas entradas todas reais, os autovalores e autovetores de  $A$  podem ser complexos.

O problema de autovalor pode ser escrito como  $Ax - \lambda x = (A - \lambda I_n)x = 0$  onde  $I_n \in \mathbb{R}^{n \times n}$  é a matriz identidade. Como  $x \neq 0$ , essa equação tem soluções apenas se

$$p(\lambda) = \det(A - \lambda I_n) = 0. \quad (1)$$

De fato, podemos mostrar que os autovalores de  $A$  são exatamente as soluções de (1) em  $\mathbb{C}$ . Observe que  $p(\lambda)$  é um polinômio de grau  $n$  na variável  $\lambda$ , então, pelo teorema fundamental da álgebra, a equação  $p(\lambda) = 0$  tem  $n$  raízes em  $\mathbb{C}$ . Assim,  $A$  tem  $n$  autovalores em  $\mathbb{C}$ . O polinômio  $p(\lambda)$  se chama polinômio característico. Observe, entretanto, que alguns autovalores podem ter o mesmo valor. Por exemplo, todos os autovalores de  $I_n$  valem 1.

Uma propriedade importante que usaremos neste exercício computacional é a seguinte: os autovalores e autovetores de matrizes simétricas são reais, isto é,  $\lambda \in \mathbb{R}$  e  $x \in \mathbb{R}^n$ .

### 1.3 Ordem de convergência

**Definição 1** (ordem de convergência). *Seja  $\mathbb{R}^n \ni x_k \rightarrow x^*$  com  $x_k \neq x^*$  para todo  $k \geq 1$ . Se existem constantes  $\eta > 0$  e  $\alpha > 0$  com*

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^\alpha} = \eta$$

*então  $x_k$  converge para  $x^*$  de ordem  $\alpha$  com erro assintótico constante  $\eta$ .*

Dois casos importantes que aparecem frequentemente são:

- Se  $\alpha = 1$  e  $\eta < 1$ , então  $x_k$  converge *linearmente*.
- Se  $\alpha = 2$ , então  $x_k$  converge *quadraticamente*.

## 2 Método das potências

### 2.1 Descrição do método

O método das potências pode ser usado quando a matriz  $A \in \mathbb{R}^{n \times n}$  tem  $n$  autovetores linearmente independentes, e os autovalores podem ser ordenados em magnitude como:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| > 0.$$

Observe que a primeira desigualdade é estrita. Quanto este ordenamento pode ser feito,  $\lambda_1$  é chamado autovalor *dominante* de  $A$ .

O método das potências é um método simples para calcular o autovalor dominante  $\lambda_1$ . Ele consiste simplesmente na iteração seguinte: dado um vetor inicial  $x_0 \in \mathbb{R}^n$ ,

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}.$$

A sequência  $x_k$  satisfaz  $x_k \rightarrow x^*$ , onde  $x^*$  é o autovetor associado a  $\lambda_1$ . A sequência

$$\mu_k = \frac{x_k^\top (Ax_k)}{x_k^\top x_k}$$

satisfaz  $\mu_k \rightarrow \lambda_1$  quando  $k \rightarrow \infty$ , então usaremos  $x_k$  como aproximação de  $x^*$  e  $\mu_k$  como aproximação de  $\lambda_1$ , para  $k$  suficientemente grande.

O método das potências produz uma sequência  $x_k$  que converge linearmente para  $x^*$  e uma sequência  $\mu_k$  que converge linearmente para  $\lambda_1$ . Em ambos casos, o erro assintótico  $\eta$  vale  $|\lambda_2/\lambda_1| < 1$ ; veja Definição 1. Se a matriz  $A$  for simétrica, o erro assintótico  $\eta$  para a convergência  $\mu_k \rightarrow \lambda_1$  vale  $|\lambda_2/\lambda_1|^2$ , então a convergência é mais rápida quando  $A$  é simétrica. Porém, o erro assintótico  $\eta$  para a convergência do autovetor  $x_k \rightarrow x^*$  continua valendo apenas  $|\lambda_2/\lambda_1|$ .

Em geral o segundo autovalor  $\lambda_2$  não é conhecido, então não podemos calcular o erro assintótico  $\eta$ . Porém, podemos calcular  $\eta$  numericamente para matrizes cujos autovalores são conhecidos; isso pode ser usado para fazer testes e verificar que nossa implementação do método é correta. Vamos dar um exemplo agora. Consideramos a matriz

$$A = \begin{pmatrix} -2 & -4 & 2 \\ -2 & 1 & 2 \\ 4 & 2 & 5 \end{pmatrix} \quad (2)$$

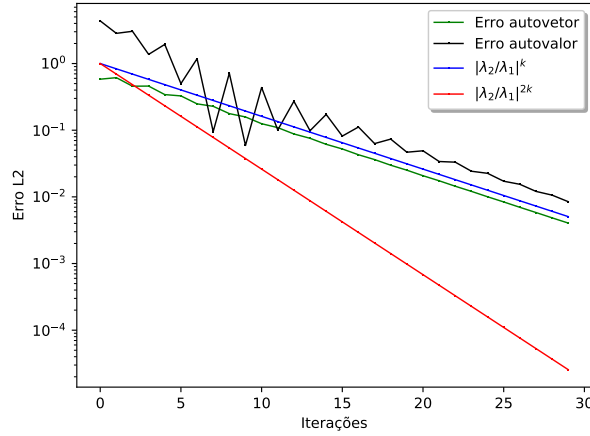


Figura 1: Uma escala logarítmica é usada para plotar os erros. Neste exemplo, observamos que os gráficos de erro de aproximação do autovalor  $|\mu_k - \lambda_1|$  (em preto), do erro de aproximação do autovetor  $\|x_k - x^*\|$  (em verde), e de  $|\lambda_2/\lambda_1|^k$  (em azul) são assintoticamente paralelos, o que significa que está verificado numericamente o erro assintótico teórico  $\eta = |\lambda_2/\lambda_1| = 5/6$  para o autovalor e o autovetor. Também observamos que não obtemos o erro assintótico teórico  $|\lambda_2/\lambda_1|^2$  (em vermelho), pois  $A$  dada por (2) não é simétrica neste exemplo.

cujos autovalores exatos são  $\lambda_1 = 6$ ,  $\lambda_2 = -5$ ,  $\lambda_3 = 3$ . Então o erro assintótico teórico é  $\eta = |\lambda_2/\lambda_1| \approx 0.8333$ . Rodamos 30 iterações do método das potências e obtemos a aproximação

$$\mu_{30} = 6.00609153.$$

O erro de aproximação do autovalor é, então,

$$|\mu_{30} - \lambda_1| = 0.00609153.$$

Observamos que este erro é de fato da mesma ordem que  $\eta^{30} = (5/6)^{30} \approx 0.00421$ . Para visualizar este erro, plotamos na Figura 1 o erro  $\|x_k - x^*\|$  para o autovetor, o erro  $|\mu_k - \lambda_1|$  para o autovalor, e a função  $\eta^k = (5/6)^k$ , onde  $k$  é o número de iterações. Quando o erro assintótico é satisfeito, os gráficos são assintoticamente paralelos.

## 2.2 Exercício 1

Para o algoritmo, precisaremos de um critério de parada. Usaremos a convergência do autovetor como critério de parada, pois a convergência do autovalor pode ser mais rápida se a matriz for simétrica. O critério de parada será, então,  $\|x_k - x^*\| \leq \varepsilon$ , onde  $\varepsilon$  pode ser escolhido como  $\varepsilon = 10^{-15}$ , o que é perto da precisão da máquina (lembre-se de que  $x^*$  é o autovetor associado a  $\lambda_1$ ). Pode acontecer que a convergência seja lenta, então também fixaremos um número máximo de iterações, chamado `itmax`. Pode escolher um valor entre 30 e 70 para `itmax`, e ajustar este valor em função do problema, lembrando que o objetivo é obter (se possível) uma aproximação da ordem de precisão da máquina. Deve-se escolher uma inicialização  $x_0 \in \mathbb{R}^n$  aleatória; pode usar funções de `numpy.random` para gerar números aleatórios.

O valor (quase) exato dos autovalores e dos autovetores pode ser obtido usando `numpy.linalg.eig`. Use esta função quando precisar para ter uma referência e para calcular numericamente a ordem de convergência, o erro assintótico, e o critério de parada.

O objetivo desta tarefa computacional é implementar o método das potências com o critério de parada, estudar a ordem de convergência e o erro assintótico numéricos, e apresentar o histórico dos erros seguindo o padrão da Figura 1. O algoritmo deve ser aplicado às duas matrizes seguintes:

1.  $A = B + B^T$ , onde  $B \in \mathbb{R}^n$  com  $n = 10$  e os coeficientes de  $B$  são coeficientes aleatórios entre 0 e 1.
2.  $A = BDB^{-1}$ , onde  $B \in \mathbb{R}^n$  é uma matriz com coeficientes aleatórios entre 0 e 1. A matriz  $D \in \mathbb{R}^n$  é uma matriz diagonal, com todos os coeficientes estritamente positivos e distintos na diagonal. Os autovalores de  $A$  são os coeficientes na diagonal de  $D$ . Pode usar a função `numpy.linalg.inv`

para calcular  $B^{-1}$ . Escolha  $n$  entre 5 e 10. Faça dois testes diferentes, o primeiro com  $\lambda_1$  e  $\lambda_2$  relativamente perto, e o segundo com  $\lambda_1$  e  $\lambda_2$  relativamente distantes. Discuta como isso afeta a velocidade de convergência e plote o histórico dos erros seguindo o padrão da Figura 1.

### 3 Método da potência inverso

#### 3.1 Descrição do método

Se  $A \in \mathbb{R}^n$  for invertível, os autovalores da matriz inversa  $A^{-1}$  são  $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_n^{-1}$ , onde  $\lambda_1, \lambda_2, \dots, \lambda_n$  são os autovalores de  $A$ . Então, se aplicarmos o método das potências a  $A^{-1}$ , obteremos  $\lambda_n^{-1}$ , pois  $\lambda_n^{-1}$  é o autovalor de maior magnitude de  $A^{-1}$ . No entanto, agora precisamos da condição

$$|\lambda_1| \geq |\lambda_3| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n| > 0,$$

onde as duas últimas desigualdades são estritas. Observe que neste caso o erro assintótico depende de  $\lambda_n$  e  $\lambda_{n-1}$ .

A iteração é, agora,

$$x_{k+1} = \frac{A^{-1}x_k}{\|A^{-1}x_k\|}.$$

A sequência  $x_k$  satisfaz  $x_k \rightarrow x^*$ , onde  $x^*$  é o autovetor de  $A^{-1}$  associado a  $\lambda_n^{-1}$ . A sequência

$$\mu_k = \frac{x_k^T (A^{-1}x_k)}{x_k^T x_k}$$

satisfaz  $\mu_k \rightarrow \lambda_n^{-1}$  quando  $k \rightarrow \infty$ , então usaremos  $x_k$  como aproximação de  $x^*$  e  $\mu_k$  como aproximação de  $\lambda_n^{-1}$ , para  $k$  suficientemente grande.

Para calcular  $x_{k+1}$ , temos duas possibilidades: podemos calcular a matriz  $A^{-1}$  explicitamente ou podemos resolver o sistema linear

$$A\tilde{x}_{k+1} = x_k. \quad (3)$$

e depois definir  $x_{k+1} = \tilde{x}_{k+1}/\|\tilde{x}_{k+1}\|$ . Para resolver o sistema linear, podemos usar um método direto como a fatoração LU, ou um método iterativo como o método SOR. A escolha do método para calcular  $x_{k+1}$  depende do tamanho e das propriedades de  $A$ . Se  $A$  não for pequena, é melhor resolver um sistema linear do que calcular  $A^{-1}$ .

Observe também que no cálculo de  $\mu_k$  precisamos calcular  $A^{-1}x_k$ , mas  $A^{-1}x_k$  já é calculado em (3), então não precisa repetir este cálculo.

#### 3.2 Exercício 2

As orientações são semelhantes às orientações do método das potências. Para resolver o sistema linear (3), você implementará o método SOR. Uma vez que o método SOR converge apenas para certos tipos de matrizes, você irá primeiro implementar o critério das linhas ou o critério de Sassenfeld (você pode implementar ambos se quiser) para verificar que o método SOR pode ser aplicado. Observe que mesmo se estes critérios não são satisfeitos, o método SOR ainda pode convergir, mas isso não é garantido. O valor do parâmetro  $\omega$  no método SOR tem que satisfazer  $1 \leq \omega < 2$  (lembre-se de que o valor  $\omega = 1$  corresponde ao método de Gauss-Seidel). Faça alguns testes e escolha um valor apropriado para  $\omega$  de acordo.

O objetivo desta tarefa computacional é implementar o método da potência inverso com um critério de parada, estudar a ordem de convergência e o erro assintótico numéricos, e apresentar os resultados como na Figura 1. O algoritmo deve ser aplicado às duas matrizes seguintes:

1.  $A = B + B^T + nI_n$  onde  $I_n$  é a matriz identidade e os coeficientes de  $B \in \mathbb{R}^n$  são coeficientes aleatórios entre 0 e 1. Escolha  $n$  entre 7 e 12.
2.  $A = BDB^{-1}$ , onde  $B \in \mathbb{R}^n$  é uma matriz com coeficientes aleatórios entre 0 e 1. A matriz  $D \in \mathbb{R}^n$  é uma matriz diagonal, com todos os coeficientes estritamente positivos e distintos na diagonal. Os autovalores de  $A$  são os coeficientes na diagonal de  $D$ . Pode usar a função `numpy.linalg.inv` para calcular  $B^{-1}$ . Escolha  $n$  entre 5 e 10. Faça dois testes diferentes, o primeiro com  $\lambda_n$  e  $\lambda_{n-1}$  relativamente perto, e o segundo com  $\lambda_n$  e  $\lambda_{n-1}$  relativamente distantes. Discuta como isso afeta a velocidade de convergência e plote os resultados.

## 4 Fatoração $QR$ de matrizes

Nos capítulos anteriores, vimos dois métodos para calcular o primeiro e o último autovalor de uma matriz, mas em certas situações é importante calcular todos os autovalores de uma matriz. Para calcular todos os autovalores de uma matriz  $A \in \mathbb{R}^{n \times n}$ , podemos usar a fatoração  $QR$  tal que  $A = QR$ , onde  $Q$  é uma matriz ortogonal e  $R$  é uma matriz triangular superior. Desenvolveremos aqui esse método de fatoração  $QR$  de matrizes  $n \times n$ , através de transformações de Householder.

### 4.1 Transformação de Householder

Dado  $v \in \mathbb{R}^n$  definimos a transformação de Householder  $H_v : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dada por  $H_v = I_n - 2 \frac{vv^T}{v \cdot v}$  (onde  $I_n$  é a identidade), que a cada  $x \in \mathbb{R}^n$  associa  $H_v x = x - 2 \frac{v \cdot x}{v \cdot v} v$ . Esta transformação determina a reflexão do vetor  $x$  em relação ao espaço  $v^\perp$ . A transformação linear  $H_v$  é ortogonal e simétrica, ou seja,  $(H_v)^{-1} = H_v^T = H_v$ . De fato, temos:

$$\begin{aligned} H_v^T &= \left( I_n - 2 \frac{vv^T}{v \cdot v} \right)^T = I_n - 2 \frac{(vv^T)^T}{v \cdot v} = H_v, \\ H_v H_v &= \left( I_n - 2 \frac{vv^T}{v \cdot v} \right) \left( I_n - 2 \frac{vv^T}{v \cdot v} \right) = I_n - 4 \frac{vv^T}{v \cdot v} + 4 \frac{(vv^T)(vv^T)}{(v \cdot v)^2} = I_n. \end{aligned}$$

Observemos ainda que uma transformação ortogonal preserva a norma de um vetor, ou seja

$$\|H_v u\|^2 = H_v u \cdot H_v u = (H_v u)^T H_v u = u^T H_v H_v u = u^T I_n u = u^T u = \|u\|^2.$$

Dados dois vetores  $x, y$  não nulos em  $\mathbb{R}^n$ , podemos definir uma transformação de Householder tal que  $H_v x = \lambda y$  com  $\lambda \in \mathbb{R}$ . Para tanto basta tomarmos  $v = x + \alpha y$ , onde  $\alpha = \pm \frac{\|x\|}{\|y\|}$  (verifique!).

**Exemplo 1.** Consideremos  $x$  e  $y$  em  $\mathbb{R}^3$ , com  $x^T = (1, 1, 0)$  e  $y^T = (0, -1, 1)$ . Definindo  $v = x + y$ , calculamos  $v^T = (1, 0, 1)$  e

$$H_v x = x - 2 \frac{v \cdot x}{v \cdot v} v = x - v = -y.$$

Note que nesse exemplo, para o cálculo de  $H_v x$ , não necessitamos da representação matricial da transformação  $H_v$ , bastando calcular os produtos escalares de  $v$  por  $x$  e de  $v$  por  $v$  e depois adicionar dois vetores. Poderíamos escrever a matriz que representa  $H_v : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  como

$$H_v = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix},$$

e então multiplicar pelo vetor  $x$ . Isto não só é desnecessário, como seria computacionalmente bem mais ineficiente (teríamos  $O(n^2)$  multiplicações na montagem de  $H_v$  e também no cálculo de  $H_v x$  ao multiplicar a matriz pelo vetor, enquanto o cálculo dos produtos internos e a soma dos vetores envolvem apenas  $O(n)$  operações).

### 4.2 Fatoração $QR$

Agora iremos mostrar como transformar uma matriz  $A \in \mathbb{R}^{n \times n}$  em uma matriz  $R \in \mathbb{R}^{n \times n}$  triangular superior, i.e., com  $R_{ij} = 0$  para  $i > j$ , através de sucessivas transformações de Householder. Ou seja, vamos definir as matrizes de transformação  $H_{v_1}, H_{v_2}, \dots, H_{v_{n-1}}$ , tais que  $H_{v_{n-1}} \dots H_{v_2} H_{v_1} A = R$  e, portanto, como as transformações de Householder são ortogonais e simétricas, teremos que  $A = QR$ , com  $Q$  ortogonal dada por

$$Q = (H_{v_{n-1}} \dots H_{v_2} H_{v_1})^T = H_{v_1} H_{v_2} \dots H_{v_{n-1}}.$$

Seja  $a_1 = (m_{1,1}, m_{2,1}, \dots, m_{n,1})^T$  a primeira coluna de  $A$  e  $e_1 = (1, 0, \dots, 0)^T$  o primeiro vetor da base canônica do  $\mathbb{R}^n$ . Definimos  $v_1$  tal que a transformação  $H_{v_1}$  leve o vetor  $a_1$  em um múltiplo de  $e_1$ . Isto é feito definindo

$$v_1 = a_1 + \delta \frac{\|a_1\|}{\|e_1\|} e_1 = a_1 + \delta \|a_1\| e_1.$$

Escolheremos o  $\delta$  nesta expressão igual ao sinal do primeiro elemento de  $a_1$ . Após a aplicação de  $H_{v_1}$  à matriz  $A$ , obtemos uma matriz da forma

$$H_{v_1}A = \begin{bmatrix} * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & * & * & \dots & * \end{bmatrix},$$

onde os  $*$  representam valores quaisquer e zera-se a primeira coluna abaixo da diagonal principal. Suponha agora que foram executadas  $i - 1$  etapas do processo e que a nova matriz  $\tilde{A}_{i-1} = H_{v_{i-1}} \dots H_{v_2} H_{v_1} A$  possua zeros abaixo da diagonal principal em suas primeiras  $i - 1$  colunas. Vamos agora definir a transformação de Householder  $H_{v_i}$  a ser aplicada a seguir. Escolhemos  $v_i = a_i + \delta_i \frac{\|a_i\|}{\|e_i\|} e_i = a_i + \delta_i \|a_i\| e_i$ , onde  $a_i = (0, \dots, 0, m_{i,i}, m_{i+1,i}, \dots, m_{n,i})^T$ ,  $m_{k,i}$ ,  $k = i, \dots, n$ , são os coeficientes abaixo da diagonal da  $i$ -ésima coluna da matriz  $\tilde{A}_{i-1}$ ,  $e_i$  é o  $i$ -ésimo elemento da base canônica do  $\mathbb{R}^n$  e  $\delta_i$  é o sinal do coeficiente  $m_{i,i}$ . Como as primeiras  $i - 1$  posições do vetor  $v_i$  serão nulas, a aplicação da transformação  $H_{v_i}$  não irá alterar as primeiras  $i - 1$  linhas da corrente matriz  $\tilde{A}_{i-1}$  e nem as suas primeiras  $i - 1$  colunas.

**Exemplo 2.** Vamos supor que após duas etapas do processo tenhamos chegado à matriz seguinte:

$$\tilde{A}_2 = \begin{bmatrix} 2 & 1 & 0 & -1 & 1 \\ 0 & 3 & 0 & 1 & 2 \\ 0 & 0 & 2 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \\ 0 & 0 & 2 & 3 & 1 \end{bmatrix}.$$

Teríamos então  $v_3 = a_3 + \|a_3\| e_3$ , com  $a_3^T = (0, 0, 2, -1, 2)$  e  $e_3^T = (0, 0, 1, 0, 0)$ . Assim,  $v_3^T = (0, 0, 5, -1, 2)$  e após a aplicação de  $H_{v_3}$  à matriz  $A$  obteríamos a nova matriz (após 3 etapas):

$$\tilde{A}_3 = \begin{bmatrix} 2 & 1 & 0 & -1 & 1 \\ 0 & 3 & 0 & 1 & 2 \\ 0 & 0 & -3 & -11/3 & 2/3 \\ 0 & 0 & 0 & 2/15 & 5/3 \\ 0 & 0 & 0 & 11/15 & 5/3 \end{bmatrix}.$$

Note neste exemplo que, nesta etapa, só alteramos a submatriz onde tanto  $i \geq 3$ , como  $j \geq 3$ , e que o resultado corresponde à aplicação de uma transformação de Householder  $H_{\tilde{v}_3}$  de  $\mathbb{R}^3$  em  $\mathbb{R}^3$  às 3 colunas desta submatriz, com  $\tilde{v}_3^T = (5, -1, 2)$  (Faça as contas!). Este é um fato geral. A aplicação da matriz  $H_{v_k}$  à matriz  $\tilde{A}_{k-1}$  na  $k$ -ésima etapa apenas altera a submatriz onde tanto  $i \geq k$ , como  $j \geq k$ . Tal fato deveria ser usado na implementação do algoritmo, evitando-se operações desnecessárias. Após as  $n - 1$  etapas do processo, a matriz  $\tilde{A}_{n-1}$  abaixo da diagonal principal estará zerada.

### 4.3 Cálculo de autovalores usando a fatoração $QR$

Dado  $A \in \mathbb{R}^{n \times n}$  e a fatoração  $A = QR$ , definimos  $A_1 = A$ ,  $R_1 = R$  e  $Q_1 = Q$ . Definimos  $A_2 = R_1 Q_1$  e procedemos a uma nova fatoração  $A_2 = Q_2 R_2$ . Repetindo este processo, obtemos uma sequência de matriz  $A_k$ . Uma vez  $A_k$  obtida, calculamos a fatoração  $A_k = Q_k R_k$ , e definimos  $A_{k+1} = R_k Q_k$ . Com essa construção obtemos o resultado seguinte.

**Teorema 1.** Seja  $A \in \mathbb{R}^{n \times n}$  com os autovalores  $\lambda_i \in \mathbb{R}$ ,  $i = 1, \dots, n$ . Suponhamos que os autovalores satisfazem

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0.$$

As matrizes  $A_k$  definidas acima convergem para uma matriz triangular superior  $U$  com entradas diagonais  $\lambda_i$ ,  $i = 1, \dots, n$ . Se  $A$  é simétrico, então  $U$  é diagonal também.

A prova deste teorema é difícil e não pode ser feita aqui. Na prática, não obtemos a matriz triangular superior  $U$ , mas paramos o algoritmo quando  $k$  é suficientemente grande, e obtemos uma aproximação  $A_k \approx U$ . Por exemplo, podemos parar o algoritmo quando todos os coeficientes de  $A_k$  abaixo da diagonal são menores que um  $\varepsilon$  pequeno, escolhido pelo usuário no início. Nesse caso, a matriz  $A_k$

é quase triangular superior, no sentido de que os coeficientes abaixo da diagonal são pequenos, e os coeficientes na diagonal de  $A_k$  são uma boa aproximação dos autovalores de  $A_k$ .

Quando os autovalores de  $A$  não satisfazem  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ , a convergência do processo descrito acima não é garantida e o método  $QR$  pode falhar. Também, quando a diferença de magnitude dos autovalores é pequena, a convergência do método  $QR$  pode ser lenta.

**Exemplo 3.** Use o método  $QR$  para calcular os autovalores de

$$A = \begin{bmatrix} 5 & -2 \\ -2 & 8 \end{bmatrix}.$$

**Solução.** O polinômio característico de  $A$  é  $p(\lambda) = \det(A - \lambda I) = (5 - \lambda)(8 - \lambda) - 4$ . As duas raízes desse polinômio de ordem 2 são 4 e 9, então os autovalores de  $A$  são 4 e 9. Agora vamos calcular as matrizes  $A_k$  do método  $QR$  e observar que elas convergem para uma matriz diagonal  $U$  com os autovalores 4 e 9 na diagonal.

**Iteração 1.**

$$A_1 = A = \begin{bmatrix} 5 & -2 \\ -2 & 8 \end{bmatrix}, \quad Q_1 = Q = \begin{bmatrix} -0.928 & 0.371 \\ 0.371 & 0.928 \end{bmatrix}, \quad R_1 = R = \begin{bmatrix} -5.385 & 4.828 \\ 0 & 6.685 \end{bmatrix}$$

**Iteração 2.**

$$A_2 = R_1 Q_1 = \begin{bmatrix} 6.793 & 2.483 \\ 2.483 & 6.207 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} -0.939 & -0.343 \\ -0.343 & 0.939 \end{bmatrix}, \quad R_2 = \begin{bmatrix} -7.233 & -4.463 \\ 0 & 4.977 \end{bmatrix}$$

**Iteração 3.**

$$A_3 = R_2 Q_2 = \begin{bmatrix} 8.325 & -1.709 \\ -1.709 & 4.675 \end{bmatrix},$$

$\vdots$

**Iteração 6.**

$$A_6 = R_5 Q_5 = \begin{bmatrix} 8.994 & 0.173 \\ 0.173 & 4.006 \end{bmatrix},$$

$\vdots$

**Iteração 12.**

$$A_{12} = R_{11} Q_{11} = \begin{bmatrix} 8.9999996 & 0.00134 \\ 0.00134 & 4.00000036 \end{bmatrix}.$$

Observamos que  $A_k$  está convergindo para a matriz diagonal

$$U = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix},$$

com as entradas na diagonal iguais aos autovalores de  $A$ .

#### 4.4 Cálculo de autovetores usando a fatoração $QR$

Na seção anterior, calculamos apenas os autovalores de uma matriz  $A$ . Usando o método  $QR$  podemos também calcular os autovetores correspondentes quando  $A$  é uma matriz simétrica. Na seção 4.3, introduzimos a matriz  $Q_k$  que vem da fatoração  $A_k = Q_k R_k$ . Agora vamos definir uma matriz  $V_k \in \mathbb{R}^{n \times n}$  definida por

$$V_k = Q_1 Q_2 Q_3 \dots Q_k = \prod_{i=1}^k Q_i.$$

Podemos mostrar que a matriz  $V_k$  é ortogonal e que  $V_k$  se aproxima de  $V$  quando  $k \rightarrow \infty$ , onde  $V$  é uma matriz ortogonal cujas colunas são os autovetores de  $A$ . A ordem dos autovetores corresponde a ordem dos autovalores na diagonal da matriz  $U$ .

Num algoritmo numérico, atualizamos  $V_k$  em cada iteração usando a propriedade  $V_k = V_{k-1} Q_k$ . Assim, para  $k$  suficientemente grande, obtemos uma aproximação dos autovetores.

## 4.5 Exercício 3

Quando as condições do Teorema 1 não são satisfeitas, o cálculo de autovalores usando a fatoração  $QR$  pode falhar. Se os autovalores forem complexos, o algoritmo não converge para os autovalores complexos. Se  $|\lambda_1| = |\lambda_2|$  a convergência pode falhar, e se  $|\lambda_1| \approx |\lambda_2|$  a convergência pode ser devagar. Neste exercício vamos ver alguns exemplos de casos onde o algoritmo não converge para os autovalores ou converge devagar.

1. Consideremos a matriz

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

Calcule o valor exato dos autovalores de  $A$  usando o polinômio característico. Calcule também o valor exato dos autovetores de  $A$  resolvendo os sistemas lineares  $(A - \lambda_k I)x = 0$  para  $k = 1, 2, 3$  (Observe que temos  $\det(A - \lambda_k I) = 0$ , pois  $x \neq 0$ , então esse sistema linear tem apenas duas equações independentes, e por isso tem uma infinidade de soluções  $x$ . Uma solução única pode ser obtida escolhendo uma normalização e escolhendo o sinal positivo para a primeira entrada do autovetor). Os autovetores  $v_k$  devem ser normalizados, i.e. eles devem satisfazer  $\|v_k\| = 1$ , e a primeira entrada do autovetor deve ser positiva (ou a segunda entrada se a primeira entrada for zero). Depois, calcule uma aproximação numérica dos autovalores e autovetores de  $A$  usando o método  $QR$ . Verifique a convergência para os autovetores e autovalores calculados explicitamente. Para a comparação, é importante que todos os autovetores sejam normalizados.

2. Consideremos a matriz

$$A = \begin{bmatrix} 1 & 1 \\ -3 & 1 \end{bmatrix}.$$

Calcule os autovalores de  $A$  usando o polinômio característico. Depois, calcule os autovalores de  $A$  usando o método  $QR$ . O que acontece? Explique brevemente o comportamento do algoritmo para essa matriz.

3. Consideremos a matriz

$$A = \begin{bmatrix} 3 & -3 \\ 0.33333 & 5 \end{bmatrix}$$

Calcule os autovalores de  $A$  usando o polinômio característico. Depois, calcule os autovalores de  $A$  usando o método  $QR$ . O que acontece? Explique brevemente o comportamento do algoritmo para essa matriz.

4. Aplica o método QR às duas matrizes do Exercício 1, e compare os valores obtidos para os autovalores (você tem que usar exatamente a mesma matriz  $B$  para fazer a comparação).

## 5 Aplicações

**Definição 2.** Um **grafo** é uma estrutura  $G = G(V, E)$ , constituída por um conjunto finito e não vazio  $V$  cujos elementos são denominados **vértices**, e um conjunto  $E$  de subconjuntos de dois elementos de  $V$  denominados **arestas**. Indicamos por  $|V|$  e  $|E|$ , respectivamente, o número de vértices e o número de arestas de  $G$ . Se  $u, v \in V$  e  $\varepsilon = \{u, v\} \in E$ , dizemos que a aresta  $\varepsilon$  incide em  $u$  e  $v$ . O **grau** de um vértice  $v$ , denotado por  $d(v)$ , é o número de arestas que incidem em  $v$ .

**Definição 3.** Seja  $G = G(V, E)$  um grafo com  $n$  vértices. A matriz de adjacência  $A(G)$  de  $G$  é a matriz quadrada de ordem  $n$  cujas entradas são  $a_{ij} = 1$  se  $\{v_i, v_j\} \in E$  para  $v_i, v_j \in V$ , e 0 nos outros casos.

A matriz  $A(G)$  é real e simétrica, formada por uns e zeros. Todos os seus autovalores são reais. **O maior autovalor  $\lambda_1(G)$  de  $A(G)$  é denominado índice de  $G$ .** Podemos mostrar que

$$d_{med} \leq \lambda_1(G) \leq d_{max}, \quad (4)$$

onde  $d_{med}$  é o grau médio dos vértices de  $G$ , e  $d_{max}$  é o grau máximo dos vértices de  $G$  (ver Definição 2).



**Exemplo 4.** O grafo da Figura 2 tem  $|V| = 12$  vértices e  $|E| = 12$  arestas. A matriz de adjacência  $A(G) \in \mathbb{R}^{12 \times 12}$  deste grafo é dada por

$$A(G) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Quando usamos grafos para modelar redes, uma questão relevante é destacar os vértices mais importantes naquele contexto, isto é, os vértices mais influentes na rede. Esta influência ou importância do vértice depende do tipo de relação modelada, representada pelas arestas do grafo, e é avaliada por meio de medidas de centralidade. Dentre as mais conhecidas estão a centralidade de grau (degree centrality), a centralidade de intermediação (betweenness centrality), e a centralidade de proximidade (closeness centrality). Todas estas são medidas não espectrais.

Em 1987, Bonacich propôs uma medida de centralidade a partir de um autovetor associado ao índice do grafo. O teorema de Perron-Frobenius garante que existe um autovetor associado ao índice de  $G$  com todas as coordenadas não negativas. Isto permite introduzir uma nova medida de centralidade de vértice de um grafo.

**Definição 4.** A **centralidade de autovetor**  $x_i^*$  do vértice  $v_i$  é a  $i$ -ésima coordenada do autovetor unitário não negativo  $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$  associado ao índice  $\lambda_1$  do grafo, ou seja,

$$x_i^* = \frac{1}{\lambda_1} \sum_{j=1}^n a_{ij} x_j,$$

onde  $a_{ij}$  são as entradas da matriz de adjacência.

## 5.1 Exercício 4

Consideremos o problema de construir um metrô numa grande cidade. São escolhidos  $n$  locais estratégicos, digamos  $v_0, v_1, \dots, v_{n-1}$ , que devem possuir estações de metrô. Estas estações, juntamente com as linhas de metrô definidas para fazerem as interligações entre elas, formam uma malha ou rede. Esta rede pode ser representada por um grafo, cujos vértices são as estações e as arestas indicam as linhas de metrô que as interligam.

Para simplificar o problema, vamos considerar estações localizadas em alguns vértices de uma malha regular, como na Figura 2. Na Figura 2, temos  $|V| = 12$  estações  $v_0, v_2, \dots, v_{11}$  divididas em 3 linhas (preta, azul e vermelha), e  $|E| = 12$  arestas. As estações  $v_4, v_7$  e  $v_8$  são estações conectando duas linhas de metrô.

Neste exercício vocês vão usar uma malha regular quadrada de referência com  $6 \times 6$  pontos. O objetivo é construir 4 linhas de metrô, cada linha composta de 5 estações e 4 arestas. Numa mesma linha, cada estação deve ser conectada a pelo menos uma estação e conectada no máximo a duas estações da mesma linha. Duas linhas diferentes podem ter no máximo uma estação em comum. Duas estações conectadas da mesma linha devem ser vizinhas na malha regular, ou vizinhas na diagonal de uma caixinha da malha regular de referência (como no exemplo da Figura 2). A rede de metrô deve ser *conectada*, isso quer dizer que começando uma viagem de qualquer estação inicial, é possível atingir qualquer outro ponto da rede (a rede da Figura 2 é conectada por exemplo). Cada rede deve incluir também o ponto em cima na esquerda da malha regular, e o ponto embaixo na direita da malha regular. O grafo terá então  $|E| = 16$  arestas, mas o número de vértices depende da conectividade da rede, pois algumas estações podem pertencer a duas ou mais linhas de metrô. As estações devem ser numeradas de cima para baixo e da esquerda para a direita, como na Figura 2 (comece a numeração dos vértices com 0 para facilitar a implementação).

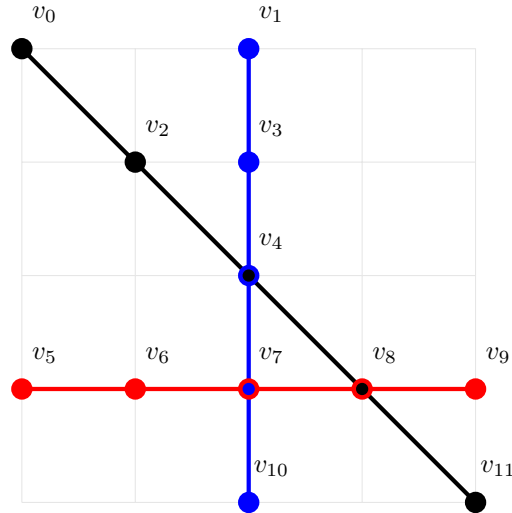


Figura 2: Exemplo de rede de metrô com  $|V| = 12$  estações  $v_0, v_1, \dots, v_{11}$  divididas em 3 linhas (preta, azul e vermelha), e  $|E| = 12$  arestas. As estações são numeradas de cima para baixo e da esquerda para a direita. O grau médio dos vértices de  $G$  neste exemplo é  $d_{med} = 24/12 = 2$ , e o grau máximo dos vértices de  $G$  é  $d_{max} = 4$  (ver Definição 2).

1. Proponha dois projetos diferentes de rede de metrô seguindo essas regras (tente propor duas redes que sejam suficientemente diferentes). Chamaremos de  $G_1$  e  $G_2$  os grafos associados a estas redes. Forneça uma figura para cada rede, procedendo como na Figura 2.
2. Construa as matrizes de adjacência  $A(G_1)$  e  $A(G_2)$ . Como a matriz de adjacência é esparsa (tem muitos zeros), uma maneira eficiente de construí-la é definir primeiro uma matriz  $\mathcal{E} \in \mathbb{R}^{|E| \times 2}$  representando o conjunto de arestas do grafo, e usar  $\mathcal{E}$  para construir a matriz de adjacência. Por exemplo, no caso da rede da Figura 2, a matriz  $\mathcal{E}$  é:

$$\mathcal{E} = \begin{pmatrix} 5 & 6 \\ 6 & 7 \\ 7 & 8 \\ 8 & 9 \\ 1 & 3 \\ 3 & 4 \\ 4 & 7 \\ 7 & 10 \\ 0 & 2 \\ 2 & 4 \\ 4 & 8 \\ 8 & 11 \end{pmatrix}.$$

Inclua as matrizes de adjacência e as matrizes de arestas  $\mathcal{E}$  no seu relatório.

3. Calcule os índices  $\lambda_1(G_1)$  e  $\lambda_1(G_2)$  dos grafos  $G_1, G_2$ . Qual das duas redes tem o maior índice? Verifique que  $\lambda_1(G_1)$  e  $\lambda_1(G_2)$  satisfazem (4).
4. Calcule as aproximações numéricas dos autovetores  $x^*(G_1), x^*(G_2)$  associados aos autovalores  $\lambda_1(G_1)$  e  $\lambda_1(G_2)$ . Para cada grafo, determine então o vértice com a maior centralidade de autovetor (ver Definição 4). Interprete o resultado.

## 6 Instruções

- O exercício deve ser feito em duplas. Apenas um aluno da dupla, o primeiro em ordem alfabética, deve entregar o exercício, destacando no relatório e no programa o nome de ambos os alunos.

- O programa deverá ser escrito em Python 3.x. e deve ser devidamente comentado e bem estruturado. A entrada e a saída deverão ser feitas de forma a ajudar o usuário a executar o programa e devem facilitar a análise dos resultados. Se o seu programa precisa de arquivos de entrada, considere que os mesmos encontram-se na mesma pasta do executável, ou faça de forma que solicite o caminho/nome do arquivo ao usuário.
- Para facilitar a correção, escreva um código para cada exercício e chame os arquivos de `ex1.py`, `ex2.py`, `ex3.py` e `ex4.py`. Vocês podem definir também um outro arquivo `metodos.py` contendo funções que são chamadas em vários exercícios.
- Será usada a biblioteca `matplotlib` para as plotagens.
- A única biblioteca que pode ser usada é `numpy` (exceto `matplotlib` para plotagem). As funções de `numpy` que calculam autovalores, autovetores, e resolvem sistemas lineares só podem ser usadas como referências para os resultados. Por exemplo, o valor exato dos autovalores pode ser obtido usando a função `numpy.linalg.eig` para calcular numericamente a ordem de convergência e o erro assintótico. Funções básicas como `numpy.dot` (para multiplicação de matrizes), `numpy.abs`, `numpy.array`, ou outras funções básicas deste tipo podem ser usadas.
- As análises e resultados obtidos devem ser organizados em um relatório que deve minimamente discutir os problemas estudados e os resultados obtidos. A entrega deverá conter um relatório (no formato .pdf), contendo a análise do problema estudado e as figuras, e os códigos usados para as simulações computacionais. A entrega também deverá ser feita em um arquivo compactado único (por exemplo, um arquivo `zip`).
- O uso de  $\text{\LaTeX}$  para escrever o relatório é incentivado. Os relatórios escritos em Latex receberão um bônus de 5% da nota final.

### Critérios de Correção

- Exercício 1 (2 pts)
- Exercício 2 (2 pts)
- Exercício 3 (2 pts)
- Exercício 4 (2 pts)
- Código bem documentado: comentários, legibilidade. (1 pt)
- Qualidade do relatório (relevância dos comentários e apresentação geral). (1 pt)
- Uso de  $\text{\LaTeX}$ (+5% da nota final)
- Será verificado se o programa entregue roda e produz saídas consistentes com os resultados apresentados no relatório.
- Em caso de atraso de até 48h, -2 pontos. Após isso, o EP não será aceito.