MINISTÉRIO DA DEFESA
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
Seção de Engenharia de Defesa (SE/10)

EE 600200: Álgebra Linear Computacional

Lista de Exercícios #1

Cap Filipe José Americano Albeny

Rio de Janeiro, RJ
Abril de 2025

1^{a} Questão

1. a) A $= \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

   Calcular o determinante para verificar se a matriz A possui inversa.

   $$det(A) = ad - bc = (1)(1) - (2)(2) = -3$$

   Logo, como

   $$det(A) \neq 0$$

   A matriz A admite inversa.

   Calculo da inversa de A:

   $$A^{-1} = \frac{1}{\det(A)} \cdot \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} \frac{-1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix}$$

2. b)

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

   Calcular o determinante para verificar se a matriz B possui inversa.

   $$det(B) = (1 \cdot 1 \cdot 4 + 2 \cdot 6 \cdot 1 + 6 \cdot 2 \cdot 1) - (6 \cdot 1 \cdot 1 + 1 \cdot 6 \cdot 1 + 4 \cdot 2 \cdot 2)$$

   Logo, como

   $$det(B) = 0$$

   A matriz B não admite inversa.

3. c) $C = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -3 & 4 & -5 & 2 \\ 1 & 1 & 0 & -2 \end{bmatrix}$

   Calcular o determinante, pela expansão dos cofatores da primeira coluna, para verificar se a matriz C possui inversa.

   $$det(C) = 1 \cdot Cof11 - 0 \cdot Cof21 - 3 \cdot Cof31 + 1 \cdot Cof41$$

$$det(C) = 1{\cdot}(-1)^2{\cdot}\begin{vmatrix} 0 & -1 & 1 \\ 4 & -5 & 2 \\ 1 & 0 & -2 \end{vmatrix} + 0{\cdot}Cof21 - 3{\cdot}(-1)^4{\cdot}\begin{vmatrix} 2 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{vmatrix} + 1{\cdot}(-1)^5{\cdot}\begin{vmatrix} 2 & 1 & 0 \\ 0 & -1 & 1 \\ 4 & -5 & 2 \end{vmatrix}$$

Cálculo das submatrizes:

$$\begin{vmatrix} 0 & -1 & 1 \\ 4 & -5 & 2 \\ 1 & 0 & -2 \end{vmatrix} = 0{\cdot}(-5){\cdot}(-2)+(-1){\cdot}2{\cdot}1+1{\cdot}4{\cdot}0-1{\cdot}(-5){\cdot}1-0{\cdot}2{\cdot}0-(-1){\cdot}4{\cdot}(-2) = -5$$

$$\begin{vmatrix} 2 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{vmatrix} = 2{\cdot}(-1){\cdot}(-2)+1{\cdot}1{\cdot}1+0{\cdot}0{\cdot}0-0{\cdot}(-1){\cdot}1-2{\cdot}1{\cdot}0-1{\cdot}0{\cdot}(-2) = 5$$

$$\begin{vmatrix} 2 & 1 & 0 \\ 0 & -1 & 1 \\ 4 & -5 & 2 \end{vmatrix} = 2{\cdot}(-1){\cdot}2+1{\cdot}1{\cdot}4+0{\cdot}0{\cdot}(-5)-0{\cdot}(-1){\cdot}4-2{\cdot}1{\cdot}(-5)-1{\cdot}0{\cdot}2 = 10$$

Resultado Final Determinante

$$\det(C) = 1 \cdot 1 \cdot (-5) + 0 - 3 \cdot 1 \cdot 5 + 1 \cdot (-1) \cdot 10$$
$$\det(C) = -5 - 15 - 10 = -30$$

Logo, como

$$det(C) = -30 \neq 0$$

A matriz C admite inversa.

Vamos criar uma matriz aumentada para o cálculo da inversa através do método de Gauss-Jordan

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ -3 & 4 & -5 & 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & -2 & 0 & 0 & 0 & 1 \end{array}\right]$$

Passo 1: Troca de linhas e operação com L_3

$$L_2 \leftrightarrow L_4 \quad \text{e} \quad L_3 \to 3L_1 + L_3$$

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & -2 & 0 & 0 & 0 & 1 \\ 0 & 10 & -2 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

Passo 2: $L_2 \to L_2 - L_1$

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & -2 & -1 & 0 & 0 & 1 \\ 0 & 10 & -2 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

Passo 3: $L_3 \to L_3 + 10L_2$

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & -2 & -1 & 0 & 0 & 1 \\ 0 & 0 & -12 & -18 & -7 & 0 & 1 & 10 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

Passo 4: $L_4 \to -12L_4 + L_3$

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & -2 & -1 & 0 & 0 & 1 \\ 0 & 0 & -12 & -18 & -7 & 0 & 1 & 10 \\ 0 & 0 & 0 & -30 & -7 & -12 & 1 & 10 \end{array}\right]$$

Passo 5: $L_2 \to -15L_2 + L_4$, $L_3 \to -5L_3 + 3L_4$

$$\left[\begin{array}{cccc|cccc} 1 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 15 & 15 & 0 & 8 & -12 & 1 & -5 \\ 0 & 0 & 60 & 0 & 14 & -36 & -2 & -20 \\ 0 & 0 & 0 & -30 & -7 & -12 & 1 & 10 \end{array}\right]$$

Passo 6: $L_1 \to -15L_1 + L_2, \; L_2 \to -4L_2 + L_3$

$$\left[\begin{array}{cccc|cccc} -15 & -15 & 0 & 0 & -7 & -12 & 1 & -5 \\ 0 & -60 & 0 & 0 & -18 & 12 & -6 & 0 \\ 0 & 0 & 60 & 0 & 14 & -36 & -2 & -20 \\ 0 & 0 & 0 & -30 & -7 & -12 & 1 & 10 \end{array}\right]$$

Passo 7: $L_1 \to -4L_1 + L_2$

$$\left[\begin{array}{cccc|cccc} 60 & 0 & 0 & 0 & 10 & 60 & -10 & 20 \\ 0 & -60 & 0 & 0 & -18 & 12 & -6 & 0 \\ 0 & 0 & 60 & 0 & 14 & -36 & -2 & -20 \\ 0 & 0 & 0 & -30 & -7 & -12 & 1 & 10 \end{array}\right]$$

Passo 8: Dividir as linhas para obter a identidade

$$L_1 \div 60, \quad L_2 \div -60, \quad L_3 \div 60, \quad L_4 \div -30$$

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & \frac{1}{6} & 1 & -\frac{1}{6} & \frac{1}{3} \\ 0 & 1 & 0 & 0 & \frac{3}{10} & -\frac{1}{5} & \frac{1}{10} & 0 \\ 0 & 0 & 1 & 0 & \frac{7}{30} & -\frac{3}{5} & -\frac{1}{30} & -\frac{1}{3} \\ 0 & 0 & 0 & 1 & \frac{7}{30} & \frac{2}{5} & -\frac{1}{30} & -\frac{1}{3} \end{array}\right]$$

Matriz Inversa Final

$$C^{-1} = \begin{bmatrix} \frac{1}{6} & 1 & -\frac{1}{6} & \frac{1}{3} \\ \frac{3}{10} & -\frac{1}{5} & \frac{1}{10} & 0 \\ \frac{7}{30} & -\frac{3}{5} & -\frac{1}{30} & -\frac{1}{3} \\ \frac{7}{30} & \frac{2}{5} & -\frac{1}{30} & -\frac{1}{3} \end{bmatrix}$$

2ª Questão

Verificando se F é um subestação de R^3 :

1. $0 \in F$
2. $u + v \in F$
3. $\alpha \cdot u \in F$

Caso w pertença ao subespaço formado por u e v:

$$a \cdot u_x + b \cdot u_y + c \cdot u_z = 0$$

$$a \cdot 1 + b \cdot 1 + c \cdot 1 = 0$$

$$a \cdot v_x + b \cdot v_y + c \cdot v_z = 0$$
$$a \cdot 1 + b \cdot (-1) + c \cdot (-1) = 0$$

Resolvendo o sistema:

$$a + b + c = 0$$
$$a - b - c = 0$$

Logo,

$$2a = 0$$
$$a = 0$$

Substituindo $a = 0$ no sistema:

$$b + c = 0$$
$$b - c = 0$$

Considerando k real não nulo:

$$b = k$$
$$c = -k$$

Exemplo: com $k = 2$, temos: $a = 0, b = 2, c = -2$

3ª Questão

a)

$$R_1 = \begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}\alpha & 0\\ 0 & \beta\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\alpha x\\ \beta y\end{bmatrix}$$

$$R_2 = \begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}\cos 30° & -\sin 30°\\ \sin 30° & \cos 30°\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\frac{\sqrt{3}}{2} & -\frac{1}{2}\\ \frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}$$

Seja a Transformação T abaixo:

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}\frac{\sqrt{3}}{2} & -\frac{1}{2}\\ \frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}\alpha x\\ \beta y\end{bmatrix} = \begin{bmatrix}\alpha\frac{\sqrt{3}}{2} - \beta\frac{1}{2}\\ \alpha\frac{1}{2} + \beta\frac{\sqrt{3}}{2}\end{bmatrix}$$

A transformação é representada pela matriz A:

$$A = \begin{bmatrix}\alpha\frac{\sqrt{3}}{2} & -\beta\frac{1}{2}\\ \alpha\frac{1}{2} & \beta\frac{\sqrt{3}}{2}\end{bmatrix}$$

b)

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}\alpha & 0\\ 0 & \beta\end{bmatrix}\begin{bmatrix}\frac{\sqrt{3}}{2} & -\frac{1}{2}\\ \frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\alpha\frac{\sqrt{3}}{2} - \alpha\frac{1}{2}\\ \beta\frac{1}{2} + \beta\frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}$$

A matriz B representa a transformação invertando a ordem das operações:

$$B = \begin{bmatrix}\alpha\frac{\sqrt{3}}{2} & -\alpha\frac{1}{2}\\ \beta\frac{1}{2} & \beta\frac{\sqrt{3}}{2}\end{bmatrix}$$

Logo, temos que a inversão das operações geram matrizes diferentes A $\neq B$.

4ª Questão

```
# Codigo disponivel em:
# https://github.com/albenyfjaa/alc-albeny

# Classe Backsolve
class backsolve:
    def __init__(self, A, b):
        self.A = A
        self.b = b
        print("A: \n", self.A)
        print("\n b: \n", self.b)

        n = len(self.A) # Armazena tamanho de n com base
            na quantidade de linhas da matriz

        # Armazenar dimensoes de A e b
        dimensao_a = A.shape
        dimensao_b = b.shape
        print("\n Dimensao A: ", dimensao_a, "Dimensao B
           : ", dimensao_b)

        # Verificar se A e uma matriz quadrada
        if dimensao_a[0] != dimensao_a[1]:
            print("\n A nao e uma matriz quadrada.
               Utilize uma matriz quadrada para A.")
            exit()

        #Verificar se existe o produto entre A e b
        if dimensao_a[1] != dimensao_b[0]:
            print("\n A e b nao possui dimensoes
               compativeis para o produto.")
            exit()

        # Verificacao se a matriz e triangular superior
        for j in range(n):
            for i in range(j + 1, n):
                if A[i][j] != 0:
                    raise ValueError("Nao e uma matriz
                       triangular superior.")

        # Verifica se algum elemento da diagonal
           principal e zero
```

```
        for i in range(n):
            if A[i][i] == 0:
                raise Exception(f"Elemento nulo na
                   diagonal principal na posicao {i+1},{
                   i+1}. Certfique-se que a diagonal
                   principal da matriz A nao possua
                   valor 0.")

        x = np.zeros(n) # Defini um valor de x qualquer
           (zero) com base no tamanho da matriz A.

        for i in range(n - 1, 0-1, -1): # Iteracao em i
           de tras para frente (inicia no maior valor de
            i)
            print(f"-------Iteracao-------")
            print("i = ", i)
            print("b = ", self.b[i])
            print("A = ", self.A[i][i])

            sum_ax = 0 # Define variavel de soma = 0
            for j in range(i + 1, n): # Nao ha calculo
               de j na primeira interacao
                print("j = ", i+ 1)

                sum_ax += self.A[i, j] * x[j]
                print(self.A[i, j], "|" ,x[j])
                print("sum_ax = ", sum_ax)


            x[i] = (self.b[i] - sum_ax) / self.A[i,i]
            print(f"x[{i+1},1] = ", x[i])
            print(x)

# Exemplo de execucao
A = np.array([
    [1, 2, 6, 0],
    [0, 1, 6, 1],
    [0, 0, 1, 1],
    [0, 0, 0, 1]
])

b = np.array([
    [1],
```

```
    [4],
    [1],
    [1]
])

backsubstitution_obj = backsolve(A, b)
```

5ª Questão

a)

```python
    # Codigo disponivel em:
    # https://github.com/albenyfjaa/alc-albeny

import numpy as np

# Funcao Posicao Efetuador Final
def posicao_efetuador(theta1_graus, theta2_graus, L1
   =20.0, L2=15.0):

    # Verifica se angulos fornecidos estao entre 0 e
        360 graus
    if not 0 <= theta1_graus <= 360:
        raise ValueError("O angulo theta1_graus deve
            estar entre 0 e 360 graus.")

    if not 0 <= theta2_graus <= 360:
        raise ValueError("O angulo theta2_graus deve
            estar entre 0 e 360 graus.")

    # Converte angulos fornecidos para radianos
    theta1 = np.radians(theta1_graus)
    theta2 = np.radians(theta2_graus)

    # Calcular os angulos e coordenadas
    theta_total = theta1 + theta2
    XU = round(L1 * np.cos(theta1) + L2 * np.cos(
       theta_total), 1)
    YU = round(L1 * np.sin(theta1) + L2 * np.sin(
       theta_total), 1)

    print("Posicao do efetuador final XU:", XU, "cm"
       , "Posicao do efetuador final YU:", YU,"cm")

    return XU, YU

# Exemplo de execucao
posicao_efetuador(90, 0)
```

b) Seja um robô planar de dois elos com juntas rotacionais. Os parâmetros do robô são:

- Comprimento do elo 1: $L_1 = 20{,}0\,\text{cm}$
- Comprimento do elo 2: $L_2 = 15{,}0\,\text{cm}$
- Ângulo da junta 1: θ_1
- Ângulo da junta 2: θ_2

O robô está no plano $\hat{X}_U$, $\hat{Y}_U$, e desejamos encontrar a matriz de transformação homogênea 3×3, que fornece a posição e orientação da ponta do robô (efetuador final).

Transformação da base até a junta 1:

$$T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & L_1 \cdot \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & L_1 \cdot \sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformação da junta 1 até a ponta do robô:

$$T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & L_2 \cdot \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & L_2 \cdot \sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicação $T = T_1 \cdot T_2$:

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

Cálculo dos elementos de T:

Primeira linha:

$$t_{11} = \cos\theta_1 \cdot \cos\theta_2 + (-\sin\theta_1) \cdot \sin\theta_2 = \cos(\theta_1 + \theta_2)$$

$$t_{12} = \cos\theta_1 \cdot (-\sin\theta_2) + (-\sin\theta_1) \cdot \cos\theta_2 = -\sin(\theta_1 + \theta_2)$$

$$t_{13} = \cos\theta_1 \cdot (L_2 \cos\theta_2) + (-\sin\theta_1) \cdot (L_2 \sin\theta_2) + L_1 \cos\theta_1 = L_2 \cos(\theta_1 + \theta_2) + L_1 \cos\theta_1$$

Segunda linha:

$$t_{21} = \sin\theta_1 \cdot \cos\theta_2 + \cos\theta_1 \cdot \sin\theta_2 = \sin(\theta_1 + \theta_2)$$

$$t_{22} = \sin\theta_1 \cdot (-\sin\theta_2) + \cos\theta_1 \cdot \cos\theta_2 = \cos(\theta_1 + \theta_2)$$

$$t_{23} = \sin\theta_1 \cdot (L_2 \cos\theta_2) + \cos\theta_1 \cdot (L_2 \sin\theta_2) + L_1 \sin\theta_1 = L_2 \sin(\theta_1 + \theta_2) + L_1 \sin\theta_1$$

Terceira linha:

$$t_{31} = 0 \quad , \quad t_{32} = 0 \quad , \quad t_{33} = 1$$

Matriz Final (com $L_1 = 20$, $L_2 = 15$):

$$T = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 15\cos(\theta_1 + \theta_2) + 20\cos\theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 15\sin(\theta_1 + \theta_2) + 20\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix}$$