Nome: Daniel Brito - Ciência da Computação - UFC Campus de Crateús

RELATÓRIO - SUMÁRIO

1 - INTRODUÇÃO

1.1 - MATRIZES

1.1.1 - DEFINIÇÕES

O crescente uso dos computadores tem feito com que a teoria das matrizes seja cada vez mais aplicada em áreas como Economia, Engenharia, Matemática, Física, dentre outras. Vejamos um exemplo. A tabela a seguir representa as notas de três alunos em uma etapa:

	Química	Inglês	Literatura	Espanhol
A	8	7	9	8
B	6	6	7	6
C	4	8	5	9

Se quisermos saber a nota do aluno **B** em Literatura, basta procurar o número que fica na segunda linha e na terceira coluna da tabela.

Vamos agora considerar uma tabela de números dispostos em linhas e colunas, como no exemplo acima, mas colocados entre parênteses ou colchetes:

$$\begin{array}{l} \text{linha} \rightarrow \\ \\ \\ \end{array} \begin{pmatrix} 8 & 7 & 9 & 8 \\ 6 & 6 & 7 & 6 \\ 4 & 8 & 5 & 9 \end{pmatrix} \quad \text{ou} \quad \begin{array}{c} \begin{bmatrix} 8 & 7 & 9 & 8 \\ 6 & 6 & 7 & 6 \\ 4 & 8 & 5 & 9 \end{bmatrix} \\ \uparrow \\ \text{coluna} \end{array}$$

Em tabelas assim dispostas, os números são os elementos. As linhas são enumeradas *de cima para baixo* e as colunas, *da esquerda para direita:*

$$\begin{array}{l} \text{1ª linha} \rightarrow \\ \text{2ª linha} \rightarrow \\ \text{3ª linha} \rightarrow \end{array} \begin{bmatrix} 1 & 4 & 7 \\ 2 & \sqrt{3} & -3 \\ 0 & 0 & 5 \end{bmatrix}$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \text{1ª coluna} & \text{2ª coluna} & \text{3ª coluna} \end{array}$$

Tabelas com **m** linhas e **n** colunas (**m** e **n** números naturais diferentes de 0) são denominadas matrizes m x n. Na tabela anterior temos, portanto, uma matriz 3 x 3. Veja mais alguns exemplos:

- $\begin{bmatrix} 2 & 3 & -1 \\ 30 & -3 & 17 \end{bmatrix}$ é uma matriz do tipo 2 x 3
- $\begin{bmatrix} 2 & -5 \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix}$ é uma matriz do tipo 2 x 2

> Notação geral

Costuma-se representar as matrizes por *letras maiúsculas* e seus elementos por *letras minúsculas*, acompanhadas por *dois índices* que indicam, respectivamente, a linha e a coluna que o elemento ocupa. Assim, uma matriz **A** do tipo m x n é representada por:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{n3} & \cdots & a_{mn} \end{bmatrix}$$

ou, abreviadamente, $A = [a_{ij}]_{m \times n}$, em que **i** e **j** representam, respectivamente, a linha e a coluna que o elemento ocupa. Por exemplo, na matriz anterior, $\mathbf{a_{23}}$ é o elemento da 2ª linha e da 3ª coluna.

Na matriz $A = \begin{bmatrix} 2 & -1 & 5 \\ 4 & \frac{1}{2} & \sqrt{2} \\ 0 & 1 & -2 \end{bmatrix}$, temos:

$$\begin{cases} a_{11} = 2, a_{12} = -1 \text{ e } a_{13} = 5 \\ a_{21} = 4, a_{22} = \frac{1}{2} \text{ e } a_{23} = \sqrt{2} \\ a_{31} = 0, a_{32} = 1 \text{ e } a_{33} = -2 \end{cases}$$

Ou na matriz B = [-1 0 2 5], temos:
$a_{11} = -1$

$a_{12} = 0$
$a_{13} = 2$
$a_{14} = 5$

> Matriz linha, matriz coluna, matriz quadrada

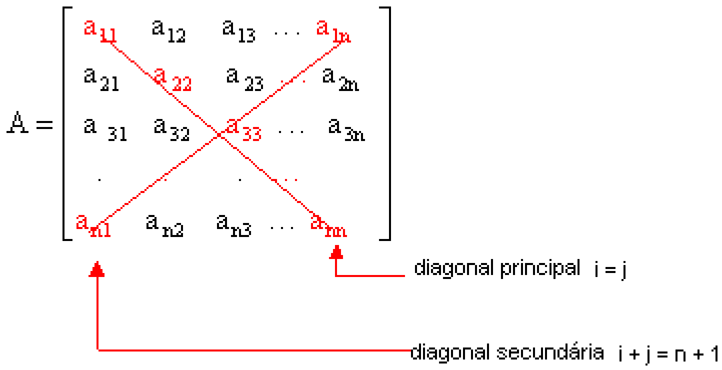
Algumas matrizes, por suas características, recebem denominações especiais.

- **Matriz linha**: matriz do tipo 1 x n, ou seja, com uma única linha. Por exemplo, a matriz A =[4 7 -3 1], do tipo 1 x 4.

- **Matriz coluna**: matriz do tipo m x 1, ou seja, com uma única coluna. Por exemplo, $B = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$, do tipo 3 x 1

- **Matriz quadrada:** matriz do tipo n x n, ou seja, com o mesmo número de linhas e colunas; dizemos que a matriz é de ordem **n**. Por exemplo, a matriz $C = \begin{bmatrix} 2 & 7 \\ 4 & 1 \end{bmatrix}$ é do tipo 2 x 2, isto é, quadrada de ordem 2.

Em uma matriz quadrada, definimos a diagonal principal e a diagonal secundária. A principal é formada pelos elementos a_{ij} tais que i = j. Na secundária, temos i + j = n + 1. Veja:



Observe a matriz a seguir:

$$A_3 = \begin{bmatrix} -1 & 2 & -5 \\ 3 & 0 & -3 \\ 5 & 7 & -6 \end{bmatrix}$$

ordem da matriz

diagonal principal

diagonal secundária

a_{11} = -1 é elemento da diagonal principal, pois i = j = 1.

a_{31}= 5 é elemento da diagonal secundária, pois
i + j = n + 1 (3 + 1 = 3 + 1).

> Matriz nula, diagonal e identidade

- **Matriz nula**: matriz em que todos os elementos são nulos; é representada por: $0_{m \times n}$.

Por exemplo, $0_{2x3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

- **Matriz diagonal**: matriz quadrada em que todos os elementos que não estão na diagonal principal são nulos. Por exemplo:

$$a)\ A_{2x2} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$b)\ B_{3x3} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

- **Matriz identidade**: matriz quadrada em que todos os elementos da diagonal principal são iguais a 1 e os demais são nulos; é representada por $\mathbf{I_n}$, sendo **n** a ordem da matriz. Por exemplo:

$$a)\ I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad b)\ I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Assim, para uma matriz identidade $I_n = \left[a_{ij}\right], a_{ij} = \begin{cases} 1, sei = j \\ 0, sei \neq j \end{cases}$.

> Matriz transposta, matriz simétrica e matriz oposta

- **Matriz transposta**: matriz A^t obtida a partir da matriz A trocando-se ordenadamente as linhas por colunas ou as colunas por linhas. Por exemplo:

$$Se\ A = \begin{bmatrix} 2 & 3 & 0 \\ -1 & -2 & 1 \end{bmatrix}, então\ A^t = \begin{bmatrix} 2 & -1 \\ 3 & -2 \\ 0 & 1 \end{bmatrix}$$

Desse modo, se a matriz **A** é do tipo m x n, $\mathbf{A}^t$ é do tipo n x m. Note que a 1ª linha de **A** corresponde à 1ª coluna de $\mathbf{A}^t$ e a 2ª linha de A corresponde à 2ª coluna de $\mathbf{A}^t$.

- **Matriz simétrica**: matriz quadrada de ordem n tal que $\mathbf{A} = \mathbf{A}^t$. Por exemplo,

$$A = \begin{bmatrix} 3 & 5 & 6 \\ 5 & 2 & 4 \\ 6 & 4 & 8 \end{bmatrix}$$

é simétrica, pois $a_{12} = a_{21} = 5$, $a_{13} = a_{31} = 6$, $a_{23} = a_{32} = 4$, ou seja, temos sempre $a_{ij} = a_{ji}$.

- **Matriz oposta**: matriz **-A** obtida a partir de **A** trocando-se o sinal de todos os elementos de **A.** Por exemplo, $Se\ A = \begin{bmatrix} 3 & 0 \\ 4 & -1 \end{bmatrix}, então - A = \begin{bmatrix} -3 & 0 \\ -4 & 1 \end{bmatrix}$.

> Igualdade de matrizes

Duas matrizes, A e B, do mesmo tipo m x n, são iguais se, e somente se, todos os elementos que ocupam a mesma posição são iguais:

$$A = B \Leftrightarrow a_{ij} = b_{ij} \; para \; \text{todo } 1 \leq i \leq m \text{ e todo } 1 \leq j \leq n$$

$$Se\ A = \begin{bmatrix} 2 & 0 \\ -1 & b \end{bmatrix},\ B = \begin{bmatrix} 2 & c \\ -1 & 3 \end{bmatrix} \text{ e } A = B, \text{ então } c = 0 \text{ e } b = 3.$$

> Adição de matrizes

Dadas as matrizes $A = \lfloor a_{ij} \rfloor_{mxn}$ e $B = \lfloor b_{ij} \rfloor_{mxn}$, chamamos de soma dessas matrizes a matriz $C = \lfloor C_{ij} \rfloor_{m \times n}$, tal que $C_{ij} = a_{ij} + b_{ij}$, para todo $1 \leq i \leq m$ e todo $1 \leq j \leq n$:

A + B = C

Exemplos:

- $\begin{bmatrix} 1 & 4 \\ 0 & 7 \end{bmatrix} + \begin{bmatrix} 2 & -1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1+2 & 4+(-1) \\ 0+0 & 7+2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 0 & 9 \end{bmatrix}$
- $\begin{bmatrix} 2 & 3 & 0 \\ 0 & 1 & -1 \end{bmatrix} + \begin{bmatrix} 3 & 1 & 1 \\ 1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 2+3 & 3+1 & 0+1 \\ 0+1 & 1+(-1) & -1+2 \end{bmatrix} = \begin{bmatrix} 5 & 4 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

Observação: A+B existe se, e somente se, **A** e **B** forem do mesmo tipo.

Propriedades:

Sendo **A**, **B** e **C** matrizes do mesmo tipo (m x n), temos as seguintes propriedades para a adição:

a) comutativa: A + B = B + A

b) associativa: (A + B) + C = A + (B + C)

c) elemento neutro: A + 0 = 0 + A = A, sendo 0 a matriz nula m x n

d) elemento oposto: A + (-A) = (-A) + A = 0

> Subtração de matrizes

Dadas as matrizes $A = [a_{ij}]_{m\times n}$ e $B = [b_{ij}]_{m\times n}$, chamamos de diferença entre essas matrizes a soma de **A** com a matriz oposta de **B**:

A - B = A + (- B)

Observe:

$$\begin{bmatrix} 3 & 0 \\ 4 & -7 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 4 & -7 \end{bmatrix} + \underbrace{\begin{bmatrix} -1 & -2 \\ 0 & 2 \end{bmatrix}}_{B} = \begin{bmatrix} 3+(-1) & 0+(-2) \\ 4+0 & -7+2 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 4 & -5 \end{bmatrix}$$

> Multiplicação de um número real por uma matriz

Dados um número real **x** e uma matriz **A** do tipo m x n, o produto de **x** por **A** é uma matriz **B** do tipo m x n obtida pela multiplicação de cada elemento de **A** por **x**, ou seja, $b_{ij} = xa_{ij}$:

B = x.A

Observe o seguinte exemplo:

$$3.\begin{bmatrix} 2 & 7 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 3.2 & 3.7 \\ 3.(-1) & 3.0 \end{bmatrix} = \begin{bmatrix} 6 & 21 \\ -3 & 0 \end{bmatrix}$$

Propriedades:

Sendo **A** e **B** matrizes do mesmo tipo (m x n) e **x** e **y** números reais quaisquer, valem as seguintes propriedades:

a) associativa: x . (yA) = (xy) . A

b) distributiva de um número real em relação à adição de matrizes: x . (A + B) = xA + xB

c) distributiva de uma matriz em relação à adição de dois números reais: (x + y) . A = xA + yA

d) elemento neutro : xA = A, para x=1, ou seja, A=A

> Multiplicação de matrizes

O produto de uma matriz por outra não é determinado por meio do produto dos seus respectivos elementos.

Assim, o produto das matrizes $A = (a_{ij})_{m \times p}$ e $B = (b_{ij})_{p \times n}$ é a matriz $C = (c_{ij})_{m \times n}$, em que cada elemento $\mathbf{c_{ij}}$ é obtido por meio da soma dos produtos dos elementos correspondentes da i-ésima linha de A pelos elementos da j-ésima coluna **B.**

Vamos multiplicar as matrizes $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ e $B = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix}$ para entender como se obtém cada elemento $\mathbf{c_{ij}}$:

- 1ª linha e 1ª coluna

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} \overset{c_{11}}{\boxed{1 \,.\, (-1) + 2 \,.\, 4}} & \\ & \end{bmatrix}$$

- 1ª linha e 2ª coluna

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 \,.\, (-1) + 2 \,.\, 4 & \overset{c_{12}}{\boxed{1.3 + 2.2}} \\ & \end{bmatrix}$$

- 2ª linha e 1ª coluna

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 \,.\, (-1) + 2 \,.\, 4 & 1.3 + 2.2 \\ \underset{c_{21}}{\boxed{3 \,.\, (-1) + 4 \,.\, 4}} & \end{bmatrix}$$

- 2ª linha e 2ª coluna

$$A = \begin{bmatrix} 1 & 2 \\ \boxed{3 \quad 4} \end{bmatrix} . \begin{bmatrix} -1 & \boxed{3} \\ 4 & \boxed{2} \end{bmatrix} = \begin{bmatrix} 1.(-1)+2.4 & 1.3+2.2 \\ 3.(-1)+4.4 & \underbrace{\boxed{3.3+4.2}}_{c_{22}} \end{bmatrix}$$

Assim, $A.B = \begin{bmatrix} 7 & 7 \\ 13 & 17 \end{bmatrix}$.

Agora observe o que aconteceria se fosse feito o contrário, ou seja, multiplicar B por A:

$$B.A = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} . \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} \boxed{(-1).1+3.3} & \boxed{(-1).2+3.4} \\ \boxed{4.1+2.3} & \boxed{4.2+2.4} \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 10 & 16 \end{bmatrix}$$

Portanto, $A.B \neq B$.A, ou seja, para a multiplicação de matrizes não vale a propriedade comutativa.

Vejamos outro exemplo com as matrizes $A = \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ -1 & 4 \end{bmatrix}$ e $B = \begin{bmatrix} 1 & 2 & 3 \\ -2 & 0 & 4 \end{bmatrix}$:

$$A.B = \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ -1 & 4 \end{bmatrix} . \begin{bmatrix} 1 & 2 & 3 \\ -2 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 2.1+3(-2) & 2.2+3.0 & 2.3+3.4 \\ 0.1+1(-2) & 0.2+1.0 & 0.3+1.4 \\ -1.1+4(-2) & -1.2+4.0 & -1.3+4.4 \end{bmatrix} = \begin{bmatrix} -4 & 4 & 18 \\ -2 & 0 & 4 \\ -9 & -2 & 13 \end{bmatrix}$$

$$B.A = \begin{bmatrix} 1 & 2 & 3 \\ -2 & 0 & 4 \end{bmatrix} . \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ -1 & 4 \end{bmatrix} = \begin{bmatrix} 1.2+2.0+3(-1) & 1.3+2.1+3.4 \\ -2.2+0.0+4(-1) & -2.3+0.1+4.4 \end{bmatrix} = \begin{bmatrix} -1 & 17 \\ -8 & 10 \end{bmatrix}$$

Da definição, temos que a matriz produto A . B só existe se o número de colunas de **A** for igual ao número de linhas de **B**:

$$A_{m \times p} \cdot B_{p \times n} = (A.B)_{m \times n}$$

A matriz produto terá o número de linhas de **A (m)** e o número de colunas de **B (n)**:

- Se $A_{3 \times 2}$ e $B_{2 \times 5}$, então $(A \cdot B)_{3 \times 5}$
- Se $A_{4 \times 1}$ e $B_{2 \times 3}$, então não existe o produto
- Se $A_{4 \times 2}$ e $B_{2 \times 1}$, então $(A \cdot B)_{4 \times 1}$

Propriedades:

Verificadas as condições de existência para a multiplicação de matrizes, valem as seguintes propriedades:

a) associativa: (A . B) . C = A . (B . C)

b) distributiva em relação à adição: A . (B + C) = A . B + A . C ou (A + B) . C = A . C + B . C

c) elemento neutro: $A \cdot I_n = I_n \cdot A = A$, sendo $\mathbf{I_n}$ a matriz identidade de ordem **n**

Vimos que a propriedade comutativa, geralmente, não vale para a multiplicação de matrizes. Não vale também o anulamento do produto, ou seja: sendo $0_{m \times n}$ uma matriz nula, $A \cdot B = 0_{m \times n}$ não implica, necessariamente, que $A = 0_{m \times n}$ ou $B = 0_{m \times n}$.

> Matriz inversa

Dada uma matriz **A**, quadrada, de ordem **n**, se existir uma matriz **A'**, de mesma ordem, tal que $A \cdot A' = A' \cdot A = I_n$, então **A'** é matriz inversa de **A** . Representamos a matriz inversa por $\mathbf{A^{-1}}$.

> Determinantes

Como já vimos, matriz quadrada é a que tem o mesmo número de linhas e de colunas (ou seja, é do tipo *n x n*). A toda matriz quadrada está associado um número ao qual damos o nome de *determinante*.

Dentre as várias aplicações dos determinantes na Matemática, temos:

- resolução de alguns tipos de sistemas de equações lineares;
- cálculo da área de um triângulo situado no plano cartesiano, quando são conhecidas as coordenadas dos seus vértices;

Determinante de 1ª ordem:

Dada uma matriz quadrada de 1ª ordem $M=[a_{11}]$, o seu determinante é o número real a_{11}:

$$\det M = |a_{11}| = a_{11}$$

Observação: representamos o determinante de uma matriz entre duas barras verticais, que não têm o significado de módulo. Por exemplo:

$M= [5] \Rightarrow \det M = 5$ ou $|5| = 5$

$M = [-3] \Rightarrow \det M = -3$ ou $|-3| = -3$

Determinante de 2ª ordem:

Dada a matriz $M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, de ordem 2, por definição o determinante associado a **M**, de 2ª ordem, é dado por:

$$\det M = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Portanto, o determinante de uma matriz de ordem 2 é dado pela diferença entre o produto dos elementos da diagonal principal e o produto dos elementos da diagonal secundária. Veja o exemplo a seguir.

$$Sendo\ M = \begin{vmatrix} 2 & 3 \\ 4 & 5 \end{vmatrix}, temos:$$

$$\det M = \begin{vmatrix} 2 & 3 \\ 4 & 5 \end{vmatrix} = \underbrace{2 \,.\, 5} - \underbrace{4 \,.\, 3} = 10 - 12 \Rightarrow \det M = -2$$

> Menor Complementar

Chamamos de *menor complementar* relativo a um elemento $\mathbf{a_{ij}}$ de uma matriz **M**, quadrada e de ordem n>1, o determinante $\mathbf{MC_{ij}}$, de ordem n - 1, associado à matriz obtida de **M** quando suprimimos a linha e a coluna que passam por $\mathbf{a_{ij}}$. Vejamos como determiná-lo pelos exemplos a seguir:

a) Dada a matriz $M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, de ordem 2, para determinar o menor complementar relativo ao elemento $\mathbf{a_{11}(MC_{11})}$, retiramos a linha 1 e a coluna 1:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \Rightarrow MC_{11} = |a_{22}| = a_{22}$$

Da mesma forma, o menor complementar relativo ao elemento a_{12} é:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \Rightarrow MC_{12} = |a_{21}| = a_{21}$$

b) Sendo $M = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, de ordem 3, temos:

$$MC_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}a_{33} - a_{23}a_{32}$$

$$MC_{12} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}.a_{33} - a_{23}.a_{31}$$

> Cofator

Chamamos de *cofator* ou *complemento algébrico* relativo a um elemento $\mathbf{a_{ij}}$ de uma matriz quadrada de ordem **n** o número $\mathbf{A_{ij}}$ tal que

$A_{ij} = (-1)^{i+j} \, . \, MC_{ij}$.

Acompanhe o exemplo:

a) Dada $M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, os cofatores relativos aos elementos $\mathbf{a_{11}}$ e $\mathbf{a_{12}}$ da matriz **M** são:

$$A_{11} = (-1)^{1+1}.\overbrace{a_{22}}^{MC_{11}} = (-1)^{2}a_{22} = +a_{22}$$

$$A_{12} = (-1)^{1+2}.\overbrace{a_{21}}^{MC_{12}} = (-1)^{3}a_{21} = -a_{21}$$

b) Sendo $M = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, vamos calcular os cofatores $\mathbf{A_{22}}$, $\mathbf{A_{23}}$ e $\mathbf{A_{31}}$:

$$A_{22} = (-1)^{2+2}.\overbrace{\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}}^{MC_{22}} = (+1)(a_{11}a_{33} - a_{13}a_{31})$$

$$A_{23} = (-1)^{2+3}.\overbrace{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}^{MC_{23}} = (-1)(a_{11}a_{32} - a_{12}a_{31})$$

$$A_{31} = (-1)^{3+1}.\overbrace{\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}}^{MC_{31}} = (+1)(a_{12}a_{23} - a_{13}a_{22})$$

1.1.2 - HISTÓRICO

O primeiro nome dado às matrizes foi por Cauchy, tableau (em português, "tabela"), mas o nome Matriz veio com James Joseph Sylvester (1814-1897), 1850. Seu significado coloquial: local onde algo se gera ou cria. Sylvester era um matemático respeitado na álgebra britânica, e foi na Universidade de Cambridge que ele conheceu o matemático inglês Arthur Cayley (1821 -1895). Sylvester via as matrizes como mero ingrediente dos determinantes, mas com Cayley elas passam a gradativamente mostrar sua importância.

O primeiro uso implícito da noção de matriz ocorreu com Lagrange que reduziu a caracterização dos máximos e mínimos, de uma função real de várias variáveis, ao estudo do sinal da forma quadrática associada à matriz das segundas derivadas dessa função. Concluímos que a Teoria das Matrizes teve como base a Teoria das Formas Quadráticas, porque seus métodos e resultados básicos foram lá gerados, mas atualmente o estudo das formas quadráticas é um mero capítulo da Teoria das Matrizes.

No estudo da álgebra linear é possível perceber que as matrizes são mais do que objetos estáticos, que gravam informações e dados, na realidade elas representam funções que agem em vetores transformando-os em outros vetores.

1.1.3 - APLICAÇÕES

As aplicações das matrizes são encontradas em todos os campos científicos.

Em física, são usadas em ramos como mecânica clássica, ótica, eletromagnetismo, mecânica quântica e eletrodinâmica quântica, além de serem essenciais na descrição do movimento de corpos rígidos.

Na teoria da probabilidade e estatística, podem ser utilizadas matrizes estocásticas que são usadas para descrever os conjuntos de probabilidades.

Em computação, as matrizes são usadas em algoritmos de rankeamento de páginas e, por exemplo, no método dos elementos finitos, em que se define um elemento e, através de matrizes, os elementos são reescritos e associados.

Além disso, as matrizes podem ser usadas em Cadeias de Markov, crescimento populacional, grafos, códigos coletores de erros, modelo quântico entre outros.

Outros exemplos:

Hiperbológrafo

Para desenhar um círculo no papel, basta um compasso; para desenhar uma elipse, bastam um pedaço de barbante e dois pregos. O desenho de uma hipérbole requer mecanismos mais complicados.

Projeto de Estrutura Metálica

Uma aplicação de Álgebra Linear à Engenharia Civil: o projeto de uma estrutura composta por vigas metálicas exige resolver um sistema de equações lineares; quanto mais complexa for esta estrutura, maior será o número de equações e de variáveis. A matriz dos coeficientes do sistema deve ser invertível para que a estrutura não colapse. Para uma mesma estrutura sujeita a forças externas variáveis, pode-se encontrar a matriz-coluna das forças que atuam sobre as vigas multiplicando-se a inversa da matriz que modela a estrutura metálica pela matriz-coluna das forças externas.

Projeto dos Eixos Traseiros de um Automóvel

Uma aplicação de Álgebra Linear à Engenharia Automobilística: obtenção da frequência natural do eixo traseiro de um automóvel através de métodos numéricos. Na indústria automobilística, hoje em dia, existe uma crescente necessidade de testes em componentes ainda na fase de projeto a fim de prever seu desempenho quando em condições de operação. Fenômenos vibratórios como a ressonância de componentes automotivos em relação às velocidades de rotação do motor e tipos de terreno devem ser levados em consideração, pois podem levar a estrutura a esforços e desgastes excessivos diminuindo sua vida útil ou aumentando o desconforto do usuário. O procedimento experimental utilizado pela indústria para testes sobre o comportamento vibracional envolve um alto custo no desenvolvimento do produto. Assim, é necessária a implantação de métodos numéricos simples e precisos de forma a predizer as frequências naturais dos componentes e a faixa de sua atuação. Para tanto, o Método das Matrizes de Transferência oferece não só rapidez e precisão, como simplicidade e versatilidade.

Mudança de Coordenadas em Sistemas de Cores

Uma aplicação de Geometria Analítica e Álgebra Linear à Computação Gráfica: o espaço espectral de cores é um espaço vetorial de dimensão 3 (correspondente às três cores primárias). Diferentes sistemas de coordenadas (conhecidos como sistemas de cores) são considerados neste espaço, de acordo com a aplicação ou o dispositivo de saída gráfica (monitor, impressora, etc.). É muitas vezes necessário passar de um sistema de coordenadas para outro, e isso é feito através de uma matriz de mudança de coordenadas. Por exemplo, a matriz de mudança de coordenadas do sistema RGB para o sistema XYZ é uma matriz 3x3 obtida quando se considera a cor branca como um ponto fixo da transformação.

Ajuste de Curvas pelo Método dos Quadrados Mínimos

Dado uma coleção de dados (pares de números) obtidos experimentalmente, busca-se uma curva que possa ser ajustada a eles de modo que a diferença entre a curva simuladora e os dados seja a menor possível. Dessa forma, predições futuras com um grau razoável de precisão podem ser feitas com base na curva obtida. Um dos métodos mais utilizados para se fazer isso é o método dos quadrados mínimos. Ele se reduz à resolução de um sistema linear cujo número de variáveis é igual ao número dos dados.

Cálculos de Curto-Circuito Trifásico

Uma aplicação de Álgebra Linear à Engenharia Elétrica: análise de circuitos elétricos sob a condição de curto-circuito através de métodos matriciais. A mesma técnica se aplica à análise estrutural de uma ponte apoiada em vários pilares e sujeita a uma carga concentrada.

Projeto de Peças de Automóveis

Atualmente, o projeto de novas peças para automóveis é realizado através de simulações em computadores, dada a necessidade de produzir modelos novos com o menor custo e em menor tempo possíveis. O método dos elementos finitos é aplicado na modelagem das peças e no estudo das tensões produzidas sobre elas para avaliar a sua resistência (procura-se reduzir ao mínimo possível a possibilidade de que uma peça se quebre ou não funcione como deva, antes de se produzir o protótipo). Isso resulta em matrizes frequentemente com milhares ou milhões de variáveis e são necessários algoritmos muito poderosos para se lidar com estas matrizes e resolver os sistemas lineares resultantes.

Método dos Elementos Finitos

O método dos elementos finitos está presente em todos os projetos industriais auxiliados por computador, modelagens e simulações numéricas. Aqui ele é aplicado no estudo da elasticidade e das deformações e tensões em placas metálicas.

1.2 - SISTEMAS LINEARES

1.2.1 - DEFINIÇÕES

> Equação Linear

Equação linear é toda equação da forma:

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n = b$$

em que $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$, **...** , $\mathbf{a_n}$ são números reais, que recebem o nome de *coeficientes* das *incógnitas* $\mathbf{x_1}$, $\mathbf{x_2}$,$\mathbf{x_3}$, **...** , $\mathbf{x_n}$, e **b** é um número real chamado *termo independente* (quando b=0, a equação recebe o nome de *linear homogênea*).

Veja alguns exemplos de equações lineares:

$3x - 2y + 4z = 7$

$-2x + 4z = 3t - y + 4$

$x + y - 3z - \sqrt{7}\,t = 0$ (homogênea)

As equações a seguir não são lineares:

$xy - 3z + t = 8$

$x^2 - 4y = 3t - 4$

$\sqrt{x} - 2y + z = 7$

> Sistema linear

Um conjunto de equações lineares da forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots a_{2n}x_n = b_2 \\ . \quad . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \quad . \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m \end{cases}$$

é um sistema linear de **m** equações e **n** incógnitas.

A solução de um sistema linear é a n-upla de números reais ordenados (r_1, r_2, r_3,..., r_n) que é, simultaneamente, solução de todas as equações do sistema.

> Matrizes associadas a um sistema linear

A um sistema linear podemos associar as seguintes matrizes:

- **matriz incompleta**: a matriz **A** formada pelos coeficientes das incógnitas do sistema.

Em relação ao sistema:

$$\begin{cases} 2x+3y-z=0 \\ 4x+y+z=7 \\ -2x+y+z=4 \end{cases}$$

a matriz incompleta é:

$$A=\begin{bmatrix} 2 & 3 & -1 \\ 4 & 1 & 1 \\ -2 & 1 & 1 \end{bmatrix}$$

- **matriz completa**: matriz **B** que se obtém acrescentando à matriz incompleta uma última coluna formada pelos termos independentes das equações do sistema.

Assim, para o mesmo sistema acima, a matriz completa é:

$$B=\begin{bmatrix} 2 & 3 & -1 & 0 \\ 4 & 1 & 1 & 7 \\ -2 & 1 & 1 & 4 \end{bmatrix}$$

> Sistemas homogêneos

Um sistema é homogêneo quando todos os termos independentes da equações são nulos:

$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=0 \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=0 \\ . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n=0 \end{cases}$$

Veja um exemplo:

$$\begin{cases} 3x - 2y + z = 0 \\ -x + 4y - 3z = 0 \\ \sqrt{2}x + 3y = 0 \end{cases}$$

A n-upla (0, 0, 0,...,0) é sempre solução de um sistema homogêneo com **n** incógnitas e recebe o nome de *solução trivial*. Quando existem, as demais soluções são chamadas não-triviais.

> Classificação de um sistema quanto ao número de soluções

Resolvendo o sistema $\begin{cases} x + y = 8 \\ 2x - y = 1 \end{cases}$, encontramos uma única solução: o par ordenado (3,5). Assim, dizemos que o sistema é **possível** (tem solução) e **determinado** (solução única).

No caso do sistema $\begin{cases} x + y = 8 \\ 2x + 2y = 16 \end{cases}$, verificamos que os pares ordenados (0,8), (1,7),(2,6),(3,5),(4,4),(5,3),...são algumas de suas infinitas soluções. Por isso, dizemos que o sistema é **possível** (tem solução) e **indeterminado** (infinitas soluções).

Para $\begin{cases} x + y = 10 \\ -x - y = 10 \end{cases}$, verificamos que nenhum par ordenado satisfaz simultaneamente as equações. Portanto, o sistema é **impossível** (não tem solução).

Resumindo, um sistema linear pode ser:

a) possível e determinado (solução única);
b) possível e indeterminado (infinitas soluções);
c) impossível (não tem solução).

> Sistema normal

Um sistema é normal quando tem o mesmo número de equações (**m**) e de incógnitas (**n**) e o determinante da matriz incompleta associada ao sistema é diferente de zero. Se m=n e det A ≠ 0, então o sistema é normal.

> Regra de Cramer

Todo sistema normal tem uma única solução dada por:

$$x_i = \frac{D_{xi}}{D}$$

em que i $\in$ { 1,2,3,...,n}, D= det A é o determinante da matriz incompleta associada ao sistema, e D_{xi} é o determinante obtido pela substituição, na matriz incompleta, da coluna **i** pela coluna formada pelos termos independentes.

> Discussão de um sistema linear

Se um sistema linear tem **n** equações e **n** incógnitas, ele pode ser:

a) **possível e determinado**, se D = detA ≠ 0; caso em que a solução é *única*.

Exemplo:

$$\begin{cases} x - y + z = 3 \\ 2x + y - z = 0 \\ 3x - y + 2z = 6 \end{cases}$$

m=n=3

$$D = \begin{vmatrix} 1 & -1 & 1 \\ 2 & 1 & -1 \\ 3 & -1 & 2 \end{vmatrix} = 3 \neq 0$$

Então, o sistema é possível e determinado, tendo solução única.

b) **possível e indeterminado**, se D= D_{x1} = D_{x2} = D_{x3} = ... = D_{xn}= 0, para n=2. Se n$\geq$3, essa condição só será válida se não houver equações com coeficientes das incógnitas respectivamente proporcionais e termos independentes não-proporcionais. Um sistema possível e indeterminado apresenta infinitas soluções.

Exemplo:

$$\begin{cases} x + 3y + 2z = 1 \\ -2x + y + z = -2 \\ -x + 4y + 3z = -1 \end{cases}$$

$D=0$, $D_x=0$, $D_y=0$ e $D_z=0$

Assim, o sistema é possível e indeterminado, tendo infinitas soluções.

c) **impossível**, se $D=0$ e $\exists D_{xi} \neq 0$, $1 \leq i \leq n$; caso em que o sistema não tem solução.

Exemplo:

$$\begin{cases} x+2y+z=1 \\ 2x+y-3z=4 \\ 3x+3y-2z=0 \end{cases}$$

$$D=\begin{vmatrix} 1 & 2 & 1 \\ 2 & 1 & -3 \\ 3 & 3 & -2 \end{vmatrix}=0 \qquad D_x=\begin{vmatrix} 1 & 2 & 1 \\ 4 & 1 & -3 \\ 0 & 3 & -2 \end{vmatrix}=35 \neq 0$$

Como $D=0$ e $D_x \neq 0$, o sistema é impossível e não apresenta solução.

> Sistemas equivalentes

Dois sistemas são equivalentes quando possuem o mesmo conjunto solução. Por exemplo, dados os sistemas:

$$S_1=\begin{cases} x+y=3 \\ 2x+3y=8 \end{cases} \quad \text{e} \quad S_2=\begin{cases} x+y=3 \\ x+2y=5 \end{cases}$$

verificamos que o par ordenado $(x, y) = (1, 2)$ satisfaz ambos e é único. Logo, S_1 e S_2 são equivalentes: $S_1 \sim S_2$.

Propriedades:

a) Trocando de posição as equações de um sistema, obtemos outro sistema equivalente. Por exemplo:

$$S_1=\begin{cases} x+y+2z=1 & (\mathrm{I}) \\ x-y=3 & (\mathrm{II}) \\ y+z=2 & (\mathrm{III}) \end{cases} \quad \text{e} \quad S_2=\begin{cases} x-y=3 & (\mathrm{II}) \\ y+z=2 & (\mathrm{III}) \\ x+y+2z=1 & (\mathrm{I}) \end{cases}$$

$S_1 \sim S_2$

b) Multiplicando uma ou mais equações de um sistema por um número **K** (K $\in$ IR*), obtemos um sistema equivalente ao anterior. Por exemplo:

$$S_1 = \begin{cases} x + 2y = 3 \ \text{(I)} \\ x - y = 0 \ \text{(II)} \end{cases} \quad \begin{matrix} \text{multiplicando a} \\ \text{equação (II) por 3} \end{matrix} \quad S_2 = \begin{cases} x + 2y = 3 \\ 3x - 3y = 0 \end{cases}$$

$S_1 \sim S_2$

c) Adicionando a uma das equações de um sistema o produto de outra equação desse mesmo sistema por um número **k** (k $\in$ IR*), obtemos um sistema equivalente ao anterior.

Por exemplo:

Dado $S_1 = \begin{cases} x + 2y = 4 \ \text{(I)} \\ x - y = 1 \ \text{(II)} \end{cases}$, substituindo a equação (II) pela soma do produto de (I) por -1 com (II), obtemos:

$$S'_1 = \frac{\begin{cases} -x - 2y = -4 \\ x - y = 1 \end{cases}}{-3y = -3} \quad S_2 = \begin{cases} x + 2y = 4 \\ -3y = -3 \end{cases}$$

$S_1 \sim S_2$, pois (x,y)=(2,1) é solução de ambos os sistemas.

> Sistemas escalonados

Utilizamos a regra de Cramer para discutir e resolver sistemas lineares em que o número de equações (**m**) é igual ao número de incógnitas (**n**). Quando **m** e **n** são maiores que três, torna-se muito trabalhoso utilizar essa regra. Por isso, usamos a técnica do *escalonamento,* que facilita a discussão e resolução de quaisquer sistemas lineares.

Dizemos que um sistema, em que existe pelo menos um coeficiente não-nulo em cada equação, está escalonado se o número de coeficientes nulos antes do primeiro coeficiente não nulo aumenta de equação para equação. Para escalonar um sistema adotamos o seguinte procedimento:

a) Fixamos como 1ª equação uma das que possuem o coeficiente da 1ª incógnita diferente de zero.

b) Utilizando as propriedades de sistemas equivalentes, anulamos todos os coeficientes da 1ª incógnita das demais equações.

c) Repetimos o processo com as demais incógnitas, até que o sistema se torne escalonado.

Vamos então aplicar a técnica do escalonamento, considerando dois tipos de sistema:

I. O número de equações é igual ao número de incógnitas (m=n)

Exemplo 1: $\begin{cases} 2x-3y-z=4 \\ x+2y+z=3 \\ 3x-y-2z=1 \end{cases}$

1ºpasso: Anulamos todos os coeficientes da 1ª incógnita a partir da 2ª equação, aplicando as propriedades dos sistemas equivalentes:

- Trocamos de posição a 1ª equação com a 2ª equação, de modo que o 1º coeficiente de x seja igual a 1.

- Trocamos a 2ª equação pela soma da 1ª equação, multiplicada por -2, com a 2ª equação:

$$\begin{cases} x+2y+z=3 \\ 2x-3y-z=4 \\ 3x-y-2z=1 \end{cases} \Leftarrow[(-2)]\Rightarrow \begin{cases} x+2y+z=3 \\ -7y-3z=-2 \\ 3x-y-2z=1 \end{cases}$$

- Trocamos a 3ª equação pela soma da 1ª equação, multiplicada por -3, com 3ª equação:

$$\begin{cases} x+2y+z=3 \\ -7y-3z=-2 \\ 3x-y-2z=1 \end{cases} \Leftarrow[(-3)]\Rightarrow \begin{cases} x+2y+z=3 \\ -7y-3z=-2 \\ -7y-5z=-8 \end{cases}$$

2º passo: Anulamos os coeficientes da 2ª incógnita a partir da 3ª equação:

- Trocamos a 3ª equação pela soma da 2ª equação, multiplicada por -1, com a 3ª equação:

$$\begin{cases} x+2y+z=3 \\ -7y-3z=-2 \\ -7y-5z=-8 \end{cases} \Leftarrow[(-1)]\Rightarrow \begin{cases} x+2y+y=3\ (\mathrm{I}) \\ -7y-3z=-2\ (\mathrm{II}) \\ -2z=-6\ (\mathrm{III}) \end{cases}$$

Agora o sistema está escalonado e podemos resolvê-lo.

$-2z=-6 \Rightarrow z=3$

Substituindo z=3 em (II):

-7y - 3(3)= -2 $\Rightarrow$-7y - 9 = -2 $\Rightarrow$y=-1

Substituindo z=3 e y=-1 em (I):

x + 2(-1) + 3= 3 $\Rightarrow$x=2

Então, x=2, y=-1 e z=3

Exemplo 2: $\begin{cases} x-2y+z=3 \\ 2x+y+z=1 \\ 3x-y+2z=2 \end{cases}$

1º passo: Anulamos todos os coeficientes da 1ª incógnita a partir da 2ª equação:

- Trocamos a 2ª equação pela soma do produto da 1ª equação por -2 com a 2ª equação:

$$\begin{cases} x-2y+z=3 \\ 2x+y+z=1 \\ 3x-y+2z=2 \end{cases} \Leftarrow[(-2)]\Rightarrow \begin{cases} x-2y+z=3 \\ 5y-z=-5 \\ 3x-y+2z=2 \end{cases}$$

- Trocamos a 3ª equação pela soma do produto da 1ª equação por -3 com a 3ª equação:

$$\begin{cases} x-2y+z=3 \\ 5y-z=-5 \\ 3x-y+2z=2 \end{cases} \Leftarrow[(-3)]\Rightarrow \begin{cases} x-2y+z=3 \\ 5y-z=-5 \\ 5y-z=-7 \end{cases}$$

2º passo: Anulamos os coeficientes da 2ª incógnita, a partir da 3ª equação:

- Trocamos a 3ª equação pela soma do produto da 2ª equação por -1 com a 3ª equação:

$$\begin{cases} x-2y+z=3 \\ 5y-z=-5 \\ 5y-z=-7 \end{cases} \Leftarrow[(-1)]\Rightarrow \begin{cases} x-2y+z=3 \\ 5y-z=-5 \\ 0z=-2 \end{cases}$$

Dessa forma, o sistema está escalonando. Como não existe valor real de z tal que 0z=-2, o sistema é impossível.

II) O número de equações é menor que o número de incógnitas (m < n)

Exemplo: $\begin{cases} x+y+z-t=6 \\ 2x+y-2z+t=-1 \\ x-2y+z+2t=-3 \end{cases}$

1º passo: Anulamos todos os coeficientes da 1ª incógnita a partir da 2ª equação:

- Trocamos a 2ª equação pela soma do produto da 1ª equação por -2 com a 2ª equação:

$$\begin{cases} x+y+z-t=6 \\ 2x+y-2z+t=-1 \Leftarrow [(-2)] \Rightarrow \\ x-2y+z+2t=-3 \end{cases} \begin{cases} x+y+z-t=6 \\ -y-4z+3t=-13 \\ x-2y+z+2t=-3 \end{cases}$$

- Trocamos a 3ª equação pela soma do produto da 1ª equação por -1 com a 3ª equação:

$$\begin{cases} x+y+z-t=6 \\ -y-4z+3t=-13 \Leftarrow [(-1)] \Rightarrow \\ x-2y+z+2t=-3 \end{cases} \begin{cases} x+y+z-t=6 \\ -y-4z+3t=-13 \\ -3y+0z+3t=-9 \end{cases}$$

2º passo: Anulamos os coeficientes da 2ª incógnita, a partir da 3ª equação:

- Trocamos a 3ª equação pela soma do produto da 2ª equação por -3 com a 3ª equação

$$\begin{cases} x+y+z-t=6 \\ -y-4z+3t=-13 \Leftarrow [(-3)] \Rightarrow \\ -3y+0z+3t=-9 \end{cases} \begin{cases} x+y+z-t=6 \\ -y-4z+3t=-13 \\ 12z-6t=30 \end{cases}$$

O sistema está escalonado. Como m<n, o sistema é possível e indeterminado, admitindo infinitas soluções. A diferença entre o número de incógnitas (n) e o de equações (m) de um sistema nessas condições é chamada *grau de indeterminação* (GI):

$$GI = n - m$$

Para resolver um sistema indeterminado, procedemos do seguinte modo:

- Consideramos o sistema em sua forma escalonada:

$$\begin{cases} x + y + z - t = 6 \\ -y - 4z + 3t = -13 \\ 12z - 6t = 30 \end{cases}$$

- Calculamos o grau de indeterminação do sistema nessas condições:

  GI = n-m = 4-3 = 1

Como o grau de indeterminação é **1**, atribuímos a uma das incógnitas um valor α, supostamente conhecido, e resolvemos o sistema em função desse valor. Sendo t=α, substituindo esse valor na 3ª equação, obtemos:

$$12z - 6\alpha = 30 \Rightarrow 12z = 30 + 6\alpha \Rightarrow Z = \frac{30 + 6\alpha}{12} = \frac{5 + \alpha}{2}$$

Conhecidos **z** e **t,** substituímos esses valores na 2ª equação:

$$-y - 4\left(\frac{5+\alpha}{2}\right) + 3\alpha = -13 \Rightarrow -y - 10 - 2\alpha + 3\alpha = -13 \Rightarrow$$

$$-y + \alpha = -13 + 10 \Rightarrow -y = -\alpha - 3 \Rightarrow y = \alpha + 3$$

Conhecidos **z,t** e **y**, substituímos esses valores na 1ª equação:

$$x + \alpha + 3 + \frac{5+\alpha}{2} - \alpha = 6 \Rightarrow 2x + 2\alpha + 6 + 5 + \alpha - 2\alpha = 12 \Rightarrow 2x +$$

$$\alpha + 11 = 12 \Rightarrow 2x = 1 - \alpha \Rightarrow x = \frac{1-\alpha}{2}$$

Assim, a solução do sistema é dada por S=$\left\{\left(\frac{1-\alpha}{2}, \alpha+3, \frac{5+\alpha}{2}, \alpha\right)\right\}$, com $\alpha \in$ IR.

Para cada valor que seja atribuído a α, encontraremos uma quádrupla que é solução para o sistema.

> Método de Gauss

O método de eliminação de Gauss para solução de sistemas de equações lineares, também conhecido como escalonamento, baseia-se em três transformações elementares, a saber:

T1 - um sistema de equações não se altera, quando permutamos as posições de duas equações quaisquer do sistema.

Exemplo: os sistemas de equações lineares

$2x + 3y = 10$

$5x - 2y = 6$

$5x - 2y = 6$

$2x + 3y = 10$

são obviamente equivalentes, ou seja, possuem o mesmo conjunto solução. Observe que apenas mudamos a ordem de apresentação das equações.

T2 - um sistema de equações não se altera, quando multiplicamos ambos os membros de qualquer uma das equações do sistema, por um número real não nulo.

Exemplo: os sistemas de equações lineares

$3x + 2y - z = 5$

$2x + y + z = 7$

$x - 2y + 3z = 1$

$3x + 2y - z = 5$

$2x + y + z = 7$

$3x - 6y + 9z = 3$

são obviamente equivalentes, pois a terceira equação foi multiplicada membro a membro por 3.

T3: um sistema de equações lineares não se altera, quando substituímos uma equação qualquer por outra obtida a partir da adição membro a membro desta equação, com outra na qual foi aplicada a transformação T2.

Exemplo: os sistemas

$15x - 3y = 22$

$5x + 2y = 32$

$15x - 3y = 22$

$\quad - 9y = - 74$

são obviamente equivalentes (ou seja, possuem o mesmo conjunto solução), pois a segunda equação foi substituída pela adição da primeira equação, com a segunda multiplicada por (-3).

> Método de Gauss-Jordan

Método para resolver sistemas de equações lineares. Baseia-se na conversão do sistema de equações lineares em uma matriz expandida (matriz na qual estão os coeficientes e os resultados). Busca-se transformar a diagonal principal em um conjunto de valores 1, enquanto os outros coeficientes são transformados em 0 (chama-se esta matriz de matriz escalonada reduzida). A vantagem deste processo é que um sistema cuja matriz aumentada é uma matriz na forma escalonada reduzida tem solução imediata, enquanto que para resolver um sistema que está apenas na forma escalonada ainda é necessário fazer uma série de substituições para obter a solução final.

Para isto, executa-se um conjunto de operações elementares sobre as linhas da matriz:

- Trocar posição entre duas linhas da matriz;
- Substituir uma equação pela mesma equação multiplicada por um escalar;
- Substituir uma equação pela soma a outra equação multiplicada por um escalar.

> Fatoração LU

A fatoração (ou decomposição) LU (em que LU vem do inglês lower e upper) é uma forma de fatoração de uma matriz não singular como o produto de uma matriz triangular inferior (lower) e uma matriz triangular superior (upper). Às vezes, se deve pré-multiplicar a matriz a ser decomposta por uma matriz de permutação. Esta decomposição se usa em análise numérica para resolver sistemas de equações (mais eficientemente) ou encontrar as matrizes inversas.

Sendo A uma matriz não singular (se o fosse, então a decomposição poderia não ser única): A = LU, onde **L** e **U** são, respectivamente, matrizes inferiores e superiores triangulares.

Para matrizes 3 x 3, isto é:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

A = PLU, ou também, PA = LU, (lembrando que as matrizes de permutação são inversíveis e a sua inversa é a sua transposta).

Onde L (Lower) e U (Upper) são, de novo, duas matrizes triangulares inferior e superior, respectivamente, e P é uma matriz de permutação.

1.2.2 - HISTÓRICO

A história dos sistemas de equações lineares começa no oriente. Em 1683, num trabalho do japonês Seki Kowa, surge a ideia de determinante (como polinômio que se associa a um quadrado de números).

O uso de determinantes no Ocidente começou dez anos depois num trabalho de Leibniz, ligado também a sistemas lineares.

A conhecida regra de Cramer é na verdade uma descoberta do escocês Colin Maclaurin (1698-1746), datando provavelmente de 1729, embora só publicada postumamente em 1748 no seu Treatise of algebra.

O suíço Gabriel Cramer (1704-1752) não aparece nesse episódio de maneira totalmente gratuita. Cramer também chegou à regra independentemente.

O francês Étienne Bézout (1730-1783), autor de textos matemáticos de sucesso em seu tempo, tratou do assunto, sendo complementado posteriormente por Laplace, em Pesquisas sobre o cálculo integral e o sistema do mundo.

O termo determinante, com o sentido atual, surgiu em 1812 num trabalho de Cauchy sobre o assunto. Neste artigo, apresentado à Academia de Ciências, sugeriu a notação que hoje é aceita como convenção.

Já o alemão Jacobi fez a leitura dessa teoria da forma como atualmente se estuda.

1.2.3 - APLICAÇÕES

Sistemas Lineares, mais precisamente, Sistemas de Equações Lineares, é ferramenta útil para a resolução de vários problemas práticos e importantes, por exemplo, problemas relacionados a tráfego de veículos, balanceamento de equações químicas, cálculo de uma alimentação diária equilibrada, circuitos elétricos e interpolação polinomial.

1.3 - ORTOGONALIZAÇÃO DE GRAM-SCHMIDT

Em matemática e análise numérica, o processo de Gram-Schmidt é um método para ortogonalização de um conjunto de vetores em um espaço com produto interno, normalmente o espaço euclidiano R^n. O processo de Gram–Schmidt recebe um conjunto finito, linearmente independente de vetores $S = \{v_1, …, v_n\}$ e retorna um conjunto ortonormal $S' = \{u_1, …, u_n\}$ que gera o mesmo subespaço S inicial.

O método leva o nome de Jørgen Pedersen Gram e Erhard Schmidt, mas pode ser encontrado antes nos trabalhos de Laplace e Cauchy. Em teoria de decomposição do grupo de Lie é generalizado pela decomposição de Iwasawa.

A aplicação do processo de Gram-Schmidt aos vetores de uma coluna matricial completa de classificação produz a fatoração QR (decomposta numa matriz ortogonal e uma matriz triangular).

> O processo de Gram-Schmidt

Define-se o operador projeção por:

$$\mathrm{proj}_{\mathbf{u}}\left(\mathbf{v}\right) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle}\mathbf{u},$$

no qual $\langle v,u \rangle$ denota o produto interno dos vetores v e u. Esse operador projeta o vetor v ortogonalmente sobre a linha gerada pelo vetor u. Se u=0, define-se $\mathrm{proj}_0(v) := 0$. i.e., o mapa projetado proj_0 é o mapa zero, enviando cada vetor ao vetor zero.

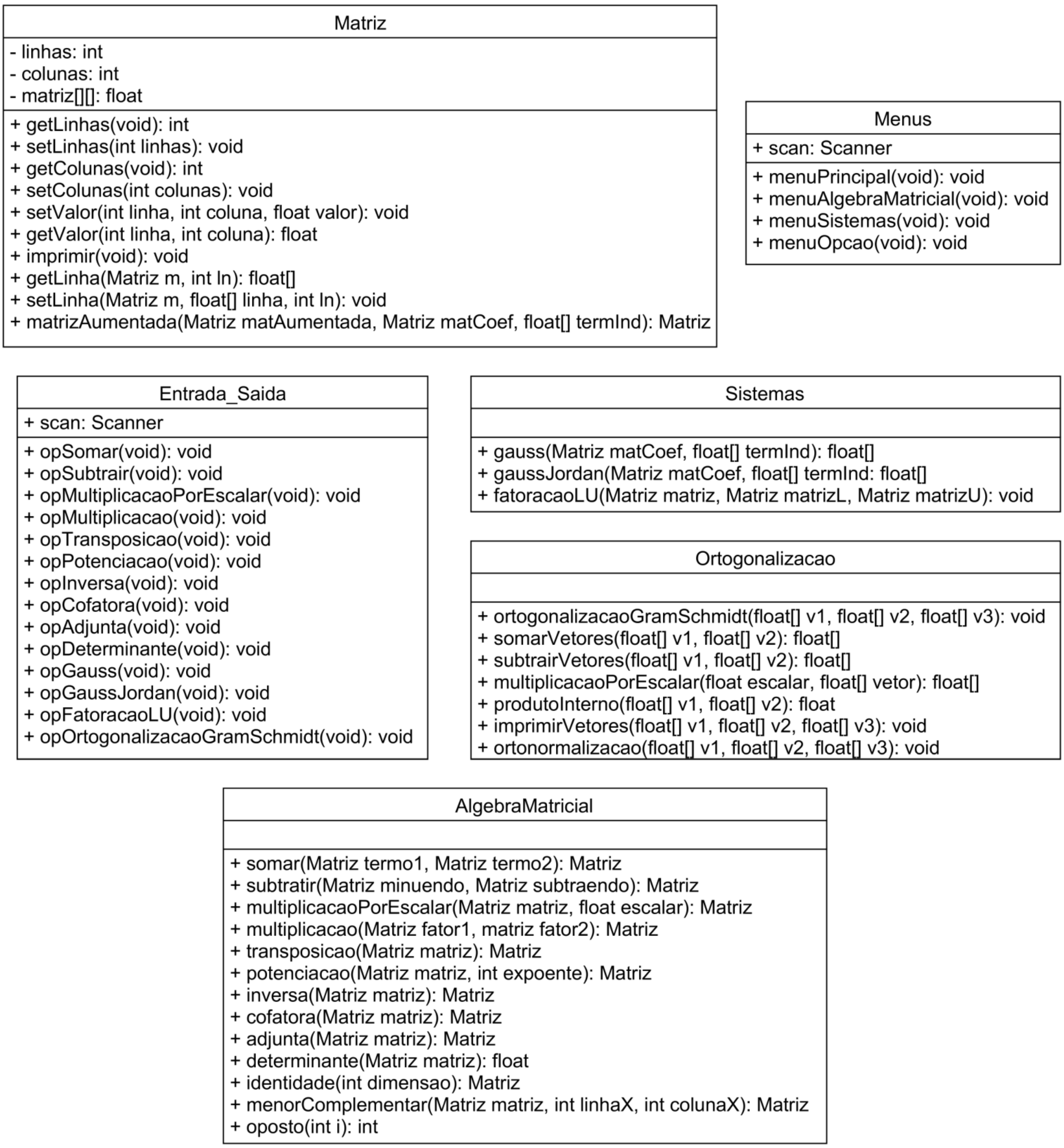
2 - GRUPO

No início, juntamente com outro colega, adaptamos alguns códigos em C de álgebra matricial elementar que havíamos implementado na disciplina de Estrutura de Dados para compor o atual programa. Mas, infelizmente, ele acabou “desistindo” da disciplina, então, a partir daí, tive que desenvolver o restante sozinho.

3 - ALGORITMO

Com base nas especificações do trabalho, foi decidido que seria mais conveniente implementar as funcionalidades do sistema utilizando a linguagem Java. Para isso, foi feito um levantamento dos requisitos e dado início a implementação do código.

3.1 - DIAGRAMA DE CLASSES



3.2 - CLASSE: Matriz

getLinhas(void): int

Retorna a quantidade de linhas de uma matriz.

setLinhas(int linhas): void

Atribui uma quantidade de linhas a uma matriz.

getColunas(void): int

Retorna a quantidade de colunas de uma matriz.

setColunas(int colunas): void

Atribui uma quantidade de colunas a uma matriz.

setValor(int linha, int coluna, float valor): void

Atribui um valor para uma determinada posição da matriz.
Matriz[linha][coluna] = 1

getValor(int linha, int coluna): float

Retorna o valor de uma determinada posição da matriz.
Matriz[linha][coluna]

imprimir(void): void

Imprime todos os elementos de uma matriz por meio do percorrimento de linhas e colunas.

getLinha(Matriz m, int ln): float[]

Retorna a linha de uma matriz.

setLinha(Matriz m, float[] linha, int ln): void

Substitui os valores da linha (ln) de uma matriz pelos valores de um vetor (linha) de mesma dimensão.

matrizAumentada(Matriz matAumentada, Matriz matCoef, float[] termInd): Matriz

Recebe uma matriz nula cuja dimensão de colunas é igual à dimensão da matriz dos coeficientes acrescida de 1. A partir daí, por meio do percorrimento de um laço de repetição, todos os valores da matriz dos coeficientes são transferidos para a matriz aumentada, bem como os respectivos termos independentes.

3.3 - CLASSE: Menus

+ menuPrincipal(void): void

Exibe as operações oferecidas pelo sistema, tais como:
1 - Álgebra Matricial
2 - Sistemas
3 - Ortogonalização

+ menuAlgebraMatricial(void): void

Exibe as operações referentes à Álgebra Matricial, tais como:
1 - Soma
2 - Subtração
3 - Multiplicação por escalar
4 - Multiplicação
5 - Transposição
6 - Potenciação
7 - Inversa
8 - Cofatora
9 - Adjunta
10 - Determinante

+ menuSistemas(void): void

Exibe os métodos de resolução de sistemas lineares, tais como:
1 - Gauss
2 - Gauss-Jordan
3 - Fatoração LU

+ menuOpcao(void): void

Exibe as opções de voltar ao Menu Principal ou Sair.

3.4 - CLASSE: Entrada_Saida

+ opSomar(void): void

Solicita ao usuário a quantidade de linhas e de colunas das matrizes que serão utilizadas na operação. Além disso, também cria uma matriz para armazenar o resultado. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, estas matrizes são enviadas para a função somar e o resultado é exibido.

+ opSubtrair(void): void

Solicita ao usuário a quantidade de linhas e de colunas das matrizes que serão utilizadas na operação. Além disso, também cria uma matriz para armazenar o resultado. Percorre os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, estas matrizes são enviadas para a função subtrair e o resultado é exibido.

+ opMultiplicacaoPorEscalar(void): void

Solicita ao usuário a quantidade de linhas e colunas da matriz, bem como um escalar que será utilizado na operação. Além disso, também cria uma matriz para armazenar o resultado. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz e o escalar são enviados para a função multiplicaçãoPorEscalar e o resultado é exibido.

+ opMultiplicacao(void): void

Solicita ao usuário a quantidade de linhas e colunas de duas matrizes. Se a operação não estiver definida para as dimensões digitadas o programa exibe uma mensagem, bem como o menuOpcao. Caso contrário, as duas matrizes são instanciadas juntamente com a matriz resultado. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, estas matrizes são enviadas para a função multiplicacao e o resultado é exibido.

+ opTransposicao(void): void

Solicita ao usuário a quantidade de linhas e colunas de uma matriz. Feito isso, duas matrizes são instanciadas, a que irá receber os valores do usuário e a que irá ser resultado da transposição. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz é enviada para a função transposicao e o resultado é exibido.

+ opPotenciacao(void): void

Solicita ao usuário a dimensão da matriz e o valor do expoente. Feito isso, duas matrizes são criadas, uma para armazenar os valores digitados pelo usuário e uma para armazenar o resultado da operação. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz e o expoente são enviados para a função potenciacao e o resultado é exibido.

+ opInversa(void): void

Solicita ao usuário a dimensão da matriz. Feito isso, duas matrizes são instanciadas, uma para armazenar os valores recebidos pelo usuário e uma para armazenar o resultado da operação. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz é enviada para a função inversa e o resultado é exibido.

+ opCofatora(void): void

Solicita ao usuário a dimensão da matriz. Feito isso, duas matrizes são instanciadas, uma para armazenar os valores recebidos pelo usuário e uma para armazenar o resultado da operação. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz é enviada para a função cofatora e o resultado é exibido.

+ opAdjunta(void): void

Solicita ao usuário a dimensão da matriz. Feito isso, duas matrizes são instanciadas, uma para armazenar os valores recebidos pelo usuário e uma para armazenar o resultado da operação. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz é enviada para a função adjunta e o resultado é exibido.

+ opDeterminante(void): void

Solicita ao usuário a dimensão da matriz. Feito isso, uma matriz é instanciada para armazenar os valores recebidos pelo usuário. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, esta matriz é enviada para a função determinante e o resultado é exibido.

+ opGauss(void): void

Solicita ao usuário a quantidade de linhas e colunas da matriz dos coeficientes. Feito isso, uma matriz é instanciada juntamente com dois vetores, um para armazenar os termos independentes e outro para armazenar a solução do sistema. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, uma matriz aumentada é instanciada, e a matriz dos coeficiente, bem como o vetor dos termos independentes são enviados para a função gauss. Por fim, o resultado é exibido.

+ opGaussJordan(void): void

Solicita ao usuário a quantidade de linhas e colunas da matriz dos coeficientes. Feito isso, uma matriz é instanciada juntamente com dois vetores, um para armazenar os termos independentes e outro para armazenar a solução do sistema. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, uma matriz aumentada é instanciada, e a matriz dos coeficiente, bem como o vetor dos termos independentes são enviados para a função gaussJordan. Por fim, o resultado é exibido.

+ opFatoracaoLU(void): void

Solicita ao usuário a quantidade de linhas e colunas da matriz dos coeficientes. Feito isso, uma matriz é instanciada juntamente com um vetor para armazenar os termos independentes. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, duas matrizes, L e U, são instanciadas para armazenar o resultado da operação ao término da execução da função fatoracaoLU. Por fim, o resultado é exibido.

+ opOrtogonalizacaoGramSchmidt(void): void

Solicita ao usuário o grau de R e a dimensão da base B. Feito isso os vetores da base são criados. Percorre-se os laços de repetição recebendo e atribuindo os valores digitados pelo usuário. Após isso, os vetores são enviados para a função ortogonalizacaoGramSchmidt e, por fim, o resultado é exibido.
Observação: A implementação da operação de ortogonalização e ortonormalização está totalmente funcional apenas em R^3 com uma base de dimensão 3.

3.5 - CLASSE: AlgebraMatricial

+ somar(Matriz termo1, Matriz termo2): Matriz

Percorre-se as matrizes por meio de dois laços de repetição acessando os valores das posições do termo1 e do termo2, estes que são somados e armazenados na respectiva posição da matriz resultado.

+ subtratir(Matriz minuendo, Matriz subtraendo): Matriz

Percorre-se as matrizes por meio de dois laços de repetição acessando os valores das posições do minuendo e do subtraendo, estes que são subtraídos e armazenados na respectiva posição da matriz resultado.

+ multiplicacaoPorEscalar(Matriz matriz, float escalar): Matriz

Percorre-se a matriz por meio de dois laços de repetição onde, a cada posição da matriz, o escalar é multiplicado pelo seu valor e o resultado é armazenado na respectiva posição da matriz resultado.

(Observação: a partir deste ponto, notei que nas especificações do trabalho pede-se o pseudocódigo apenas das operações mais importantes, não de todas. No caso de álgebra matricial, penso que as funções mais importantes são a do **determinante** e a do **menor complementar**, pois todas as outras são baseadas em operações triviais ou utilizam estas duas funções na obtenção do resultado.)

+ multiplicacao(Matriz fator1, matriz fator2): Matriz

+ transposicao(Matriz matriz): Matriz

+ potenciacao(Matriz matriz, int expoente): Matriz

+ inversa(Matriz matriz): Matriz

+ cofatora(Matriz matriz): Matriz

+ adjunta(Matriz matriz): Matriz

+ oposto(int i): int

+ identidade(int dimensao): Matriz

+ determinante(Matriz matriz): float

Esta função recebe um objeto "matriz" do tipo Matriz como parâmetro e retorna o número real (float) "det".

```
INICIO determinante

        VARIAVEIS
        i, tam: INTEIRO
        det: REAL

        tam <- getLinhas() {quantidade de linhas/colunas da matriz quadrada}

        SE tam = 1

                RETORNE matriz[0][0]

        SE tam = 2

                RETORNE matriz[0][0] * matriz[1,1] - matriz[0][1] * matriz[1][0]

        det <- 0

        PARA i de 0 ATÉ tam PASSO 1

                {chamada recursiva para matrizes de ordem acima de 2 baseando-se no teorema:}
                det <- det + (-1)^i * matriz[0][i] * determinante(menorComplementar(matriz, 0, i);

        FIM_PARA

        RETORNE det

FIM_determinante
```

+ menorComplementar(Matriz matriz, int linhaX, int colunaX): Matriz

Esta função recebe um objeto "matriz" do tipo Matriz, um inteiro linhaX e um inteiro colunaX, os quais são utilizados para suprimir a linha e a coluna de "matriz" para, no final, retornar a matriz "menor", cuja ordem é decrescida em 1 com relação à "matriz".

```
INICIO menorComplementar

        VARIAVEIS
        menor: MATRIZ
        linha, coluna, tam, i, j: INTEIRO

        linha <- -1
        ln <- getLinhhas(); {quantidade de linhas da matriz}
        cl <- getColunas(); {quantidade de colunas da matriz}

        {percorre as posições da matriz, ignorando linhas e colunas
        definidas, e atribui os outros valores à matriz menor:}

        PARA i de 0 ATÉ ln FAÇA PASSO 1

                SE i=linhaX

                        CONTINUE {ignora linha em questão, ou seja, a suprime}

                linha <- linha + 1
                coluna <- -1

                PARA j de 0 ATÉ cl FAÇA PASSO 1

                        SE j==colunaX

                                CONTINUE {ignora coluna em questão, ou seja, a suprime}

                        menor[linha][coluna+1] <- matriz[i][j]

                FIM_PARA

        FIM_PARA

        RETORNE menor

FIM_menorComplementar
```

3.6 - CLASSE: Sistemas

+ gauss(Matriz matCoef, float[] termInd): float[]

Esta função recebe um objeto “matCoef” do tipo Matriz, representando a matriz dos coeficientes, e o vetor “termInd” do tipo real (float) que contém os termos independentes do sistema linear. Após as operações, um outro vetor contendo a solução é retornado.

```
INICIO gauss

        VARIAVEIS
        n, p, max, i, j: INTEIRO
        temp, solucao: REAL[]
        t, soma, a: REAL

        n <- termInd.length {n recebe o tamanho do vetor dos termos independentes}

        PARA p=0 ATÉ p<n FAÇA PASSO 1

                {encontra a linha pivô e realiza a troca:}

                max <- p;

                PARA i=p+1 ATÉ i<n FAÇA PASSO 1

                        SE matrizCoef[i][p] > matrizCoef[max][p]

                                max <- i;

                FIM_PARA

                {temp recebe a linha p de matCoef:}
                temp <- getLinha(matrizCoef, p)
                {a linha da posição p de matrizCoef recebe a linha da posição max de matrizCoef:}
                setLinha(matCoef, getLinha(matCoef, max), p)
                {a linha da posição max de matCoef recebe temp:}
                setLinha(matCoef, temp, max)

                t <- termInd[p]
                termInd[p] <- termInd[max]
                termInd[max] <- t

                PARA i=p+1 ATÉ i<n FAÇA PASSO 1

                        a <- matrizCoef[i][p]/matrizCoef[p][p]
                        termInd[i] <- termInd - a * termInd[p]

                        PARA j=p ATÉ j<n FAÇA PASSO 1

                                matrizCoef[i][j] <- matrizCoef[i][j] - (a * matrizCoef[p][j])

                        FIM_PARA

                FIM_PARA

        FIM_PARA

        {substituição reversa, atribundo os devidos resultados ao vetor solução:}

        PARA i=n-1 ATÉ i>=0 FAÇA PASSO -1

                soma <- 0

                PARA j=i+1 ATÉ j<n FAÇA PASSO 1

                        soma <- soma + matrizCoef[i][j]*solucao[j]

                FIM_PARA

                solucao <- termInd[i]-soma/matrizCoef[i][i]

        FIM_PARA

        RETORNE solucao

FIM gauss
```

+ gaussJordan(Matriz matCoef, float[] termInd: float[]

Esta função recebe um objeto “matCoef” do tipo Matriz, representando a matriz dos coeficientes, e o vetor “termInd” do tipo real (float) que contém os termos independentes do sistema linear. Após as operações, o vetor de termos independentes é retornado, visto que no método de Gauss-Jordan o resultado do sistema é imediato.

```
INICIO gaussJordan

        VARIAVEIS
        i, j, k, m, r, s, t, ln, cl: INTEIRO
        temp, tempPivo: REAL
        tempLinha: REAL[]
        indice: REAL[][]

        s <- 0
        t <- 0
        ln <- getLinhas() {ln recebe a quantidade de linhas da matriz dos coeficientes}
        cl <- getColunas() {cl recebe a quantidade de colunas da matriz dos coeficientes}

        {Determinar o elemento pivô para a coluna atual e reorganizar as linhas:}

        PARA r DE s+1 ATÉ t<ln FAÇA PASSO 1

                SE matrizCoef[r][s] > matrizCoef[t][s]

                        t <- r

                FIM_SE

        FIM_PARA

        SE t!= s

                {tempLinha recebe a linha na posição s de matrizCoef:}
                tempLinha <- getLinha(matrizCoef, s)
                {a linha na posição s de matrizCoef recebe a linha na posição t de matrizCoef:}
                setLinha(matrizCoef, getLinha(matrizCoef, t), s)
                {a linha na posição t de matrizCoef recebe tempLinha}
                setLinha(matrizCoef, tempLinha, t)

                tempPivo <- termInd[s]
                termoInd[s] = termInd[t]
                termInd[t] = tempPivo;

        FIM_SE

        {Executa a eliminação linha por linha. Primeiro dividindo a linha atual e o elemento
        termInd por matrizCoef[i][i]:}

        PARA i DE 0 ATÉ i<ln FAÇA PASSO 1

                temp <- matrizCoef[i][i]

                PARA j DE 0 ATÉ j<cl FAÇA PASSO 1

                        matrizCoef[i][j] <- matrizCoef[i][j]/temp

                FIM_PARA

                termInd[i] = termInd[i]/temp
                matrizCoef[i][i] <- temp
```

```
        {Reduz para a matriz identidade outras linhas pela subtração de um múltiplo da linha atual.
     Não reduz a linha atual. Como cada coluna de matrizCoef é reduzida os elementos são
     substituidos pela matriz inversa de matrizCoef:}

        PARA k de 0 ATÉ k<ln FAÇA PASSO 1

                SE k!=i

                        temp <- matrizCoef[k][i]

                        PARA j de 0 ATÉ j<cl FAÇA PASSO 1

                                matrizCoef[k][j] <- matrizCoef[k][j] - temp * matrizCoef[i][j]

                        FIM_PARA

                        termInd[k] <- termInd[k] - temp * termInd[i]
                        matrizCoef[k][i] <- matrizCoef[k][i] - temp * matrizCoef[i][i]

                FIM_SE

        FIM_PARA

    FIM_PARA

    {Reordena a matriz inversa matrizCoef. As colunas são trocadas na ordem inversa que as
     linhas foram durante o pivoteamento:}

    PARA j DE cl-1 ATÉ j>=0 FAÇA PASSO -1

        k <- indice[j][0];
        m <- indice[j][1];

        SE k!=m

            PARA i DE 0 ATÉ i<ln FAÇA PASSO 1

                temp = matrizCoef[i][m]
                matrizCoef[i][m] <- matrizCoef[i][k]
                matrizCoef[i][k] <- temp

            FIM_PARA

        FIM_SE

    FIM_PARA

    RETORNE termInd

FIM gaussJordan
```

+ fatoracaoLU(Matriz matriz, Matriz matrizL, Matriz matrizU): void

Esta função recebe três objetos do tipo Matriz por parâmetro, a "matriz" representando a matriz dos Coeficientes, e a "matrizL" e "matrizU", ambas nulas, onde serão atribuídos os devidos resultados das operações.

```
INICIO fatoracaoLU

        VARIAVEIS
        soma: REAL
        i, k, j: INTEIRO

        ln <- getLinhas() {ln recebe a quantidade de linhas de matriz}
        cl <- getColunas() {cl recebe a quantidade de colunas de matriz}

        {Decompor a matriz em matrizes triangulares superior (U) e inferior (L):}

        PARA i DE 0 ATÉ i<ln FAÇA PASSO 1

                {Obtendo a matriz triangular superior:}

                PARA k DE i ATÉ k<ln FAÇA PASSO 1

                        soma <- 0

                        PARA j DE 0 ATÉ j<i FAÇA PASSO 1

                                soma <- soma + matrizL[i][j] * matriU[j][k]

                        FIM_PARA

                        matrizU[i][k] <- matriz[i][k]-soma

                FIM_PARA

                {Obtendo a matriz triangular inferior:}

                PARA k DE i ATÉ k<ln FAÇA PASSO 1

                        SE i=k

                                matrizL[i][i] <- 1 {Diagonal como 1}

                        SENÃO

                                soma <- 0

                                PARA j DE 0 ATÉ j<i FAÇA PASSO 1

                                        soma <- soma + matrizL[k][j] * matrizU[j][i]

                                FIM_PARA

                                matrizL[k][i] <- (matriz[k][i]-soma)/matrizU[i][i]

                        FIM_SE

                FIM_PARA

        FIM_PARA

FIM fatoracaoLU
```

3.7 - CLASSE: Ortogonalizacao

Não devido à complexidade do problema, mas por uma questão de falta de tempo hábil, não foi possível implementar o processo de Gram-Schmidt de forma totalmente funcional. O algoritmo desenvolvido para este trabalho se resume em ortogonalizar e ortonormalizar bases de dimensão 3 com vetores de R^3.

Baseado em um exemplo feito em sala, todas as funções necessárias para este processo foram desenvolvidas, como cálculo do produto interno, multiplicação por escalar, soma e subtração de vetores etc. No entanto, por hora, elas se aplicam somente à entrada de dados que atendem as características citadas anteriormente.

4 - CONCLUSÃO

> Objetivo

O objetivo deste trabalho foi estimular a análise do funcionamento dos processos que são realizados em operações de álgebra matricial e em sistemas lineares para, assim, nos tornarmos aptos a implementar soluções algorítmicas que resolvam problemas relacionados ao campo da álgebra linear.

> Dificuldades

A maior dificuldade na resolução deste trabalho foi a implementação dos métodos de resolução de sistemas lineares, como a Eliminação Gaussiana e o Método de Gauss-Jordan. Após muitos resultados incorretos, o debug do código feito de forma manual foi imprescindível para correção dos erros. Ainda relacionado a esta questão, outro fator que dificultou bastante foi a sobrecarga de trabalho, que acabou prejudicando um pouco o meu rendimento na disciplina, e ainda não possibilitou que eu o concluísse.

> Aprendizado

Vi neste trabalho uma ótima oportunidade para tentar conhecer os recursos de manipulação de interface gráfica oferecidos pela linguagem Java. No entanto, pelas dificuldades citadas acima, também não consegui estudar tais ferramentas e implementá-las no sistema, assim, por hora, a entrada e saída dos dados são realizadas diretamente no console da IDE.

No final, apesar das intempéries, realizar este trabalho foi muito importante para saber onde toda esta teoria é aplicada.

5 - REFERÊNCIAS

SÓ MATEMÁTICA. Matrizes. Disponível em: <https://www.somatematica.com.br/emedio/matrizes/matrizes.php>. Acesso em: 23 de novembro de 2018

UFMG. Aplicações práticas da álgebra linear. Disponível em: <http://www.mat.ufmg.br/gaal/aplicacoes/aplicacoes.html>. Acesso em: 23 de novembro de 2018

WIKIPÉDIA. Matriz (Matemática). Disponível em: <https://pt.wikipedia.org/wiki/Matriz_(matem%C3%A1tica)#Introdu%C3%A7%C3%A3o_Hist%C3%B3rica>. Acesso em: 23 de novembro de 2018

SÓ MATEMÁTICA. Sistemas. Disponível em: <https://www.somatematica.com.br/emedio/sistemas/sistemas.php>. Acesso em: 23 de novembro de 2018

MARQUES, Paulo. Sistemas Lineares, Regra de Cramer. Disponível em: <https://www.algosobre.com.br/matematica/sistemas-lineares-metodo-de-eliminacao-de-gauss.html>. Acesso em: 23 de novembro de 2018

REVISTA BW. Álgebra Linear: Método Gauss-Jordan. Disponível em: <http://www.revistabw.com.br/revistabw/metodo-gauss-jordan/>. Acesso em: 23 de novembro de 2018

WIKIPÉDIA. Sistemas de equações Lineares. Disponível em: <https://pt.wikipedia.org/wiki/Sistema_de_equa%C3%A7%C3%B5es_lineares#Hist%C3%B3rico>. Acesso em: 23 de novembro de 2018

WIKIPÉDIA. Decomposição LU. Disponível em: <https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_LU>. Acesso em: 23 de novembro de 2018

WIKIPÉDIA. Processo de Gram-Schmidt. Disponível em: <https://pt.wikipedia.org/wiki/Processo_de_Gram-Schmidt>. Acesso em: 23 de novembro de 2018