

Programação para Engenharia II

Estruturas Homogêneas

Sumário

1 Vetor

2 Manipulação de Vetor

3 Matriz

4 Manipulação de Matrizes

5 Funções Matriciais

6 Bibliografia

Vetor

Para criar um vetor¹ com os elementos em uma linha, separa-se os elementos com uma vírgula ou simplesmente um espaços.

```
1 vetorA = [1 2 3]
2 vetorB = [1,2,3]
```

Código 1: Exemplo de declaração de vetores.

¹Um vetor é uma estrutura de dados unidimensional *ou parte de uma matriz* que armazena uma sequência ordenada de elementos do mesmo tipo, podendo fazer parte de estruturas multidimensionais. 3/30

```
1 v = [1, 2, 3, 4, 5];  
2 v2 = [1; 2; 3; 4; 5];  
3  
4 soma = sum(v);           % soma dos elementos  
5 media = mean(v);         % média dos elementos
```

Código 2: Exemplo de manipulação de vetores.

Vetores podem ser unidos em um novo vetor usando colchetes. Exemplo:

$$C = [A, B]$$

Cria um vetor linha concatenando A e B.

```
1 vetorA = [1 2 3]
2 vetorB = [4 5 6]
3 VetorC = [vetorA, vetorB]
4 % O vetor resultando é [1 2 3 4 5 6]
```

Código 3: Exemplo de concatenação de vetores.

- › `vetor = 1:1:10` cria um vetor com valores de 1 até 10.
- › O primeiro número (1) é o **início** do vetor.
- › O segundo número (1) é o **intervalo** entre os elementos consecutivos.
- › O terceiro número (10) é o **fim** do vetor.
- › Resultado: [1 2 3 4 5 6 7 8 9 10].

Manipulação de Vetor

- › Para acessar um elemento específico do vetor, use `vetor(indice)`.
 - › Exemplo: `vetor = 1:1:10;`
 - › `vetor(3)` retorna o terceiro elemento (3).
- › Para modificar um elemento, atribua um novo valor à posição desejada:
 - › `vetor(3) = 15;` agora o terceiro elemento é 15.
- › É possível acessar múltiplos elementos usando vetores de índices:
 - › `vetor([2,4,6])` retorna os elementos nas posições 2, 4 e 6.

MATLAB fornece funções prontas para encontrar o maior ou menor elemento de um vetor:

- `max(A)`: retorna o maior elemento de A.
- `min(A)`: retorna o menor elemento de A.

```
1 vetor = [1 2 3 4 5]
2 max(vetor) %5
3 min(vetor) %1
```

Código 4: Exemplo de uso de `max(A)` e `min(A)`.

Matriz

Para criar uma matriz, separa-se as linhas com ponto e vírgula.

```
1 matrizA = [1 2 3; 4 5 6]
2 matrizB = [1,2,3; 4,5,6]
```

Código 5: Exemplo de declaração de matrizes.

Os números antes do ponto e vírgula pertencem à primeira linha e os números após o ponto e vírgula pertencem à segunda linha

- › **ones**: cria uma matriz ou vetor preenchido com valores iguais a 1.
- › **zeros**: cria uma matriz ou vetor preenchido com valores iguais a 0.
- › **rand**: cria uma matriz ou vetor com números reais aleatórios no intervalo $[0,1]$.
- › **randi**: cria uma matriz ou vetor com números inteiros aleatórios dentro de um intervalo definido.

Atenção!

Observação: Ao utilizar essas funções, o **primeiro argumento** indica o número de linhas e o **segundo argumento** indica o número de colunas da matriz ou vetor criado.

Manipulação de Matrizes

```
1 A = [1, 2, 3; 4, 5, 6; 7, 8, 9];  
2  
3 linhas = size(A,1);      % número de linhas  
4 colunas = size(A,2);     % número de colunas
```

Código 6: Exemplo de criação e manipulação de matrizes.

- Para acessar um elemento específico da matriz, use `matriz(linha, coluna)`.
 - » Exemplo: `matriz = [1,2,3; 4,5,6; 7,8,9]`;
 - » `matriz(2,3)` retorna o elemento da segunda linha e terceira coluna (6).
- Para modificar um elemento, atribua um novo valor à posição desejada:
 - » `matriz(2,3) = 10`; agora o elemento da segunda linha e terceira coluna é 10.
- Para acessar uma linha inteira: `matriz(2,:)` retorna todos os elementos da segunda linha.
- Para acessar uma coluna inteira: `matriz(:,3)` retorna todos os elementos da terceira coluna.
- Para acessar múltiplas linhas ou colunas específicas: `matriz([1,3],[2,3])` retorna os elementos das linhas 1 e 3 e colunas 2 e 3.


```
1 % Acesso a elementos
2 A = [10, 20, 30; 40, 50, 60];
3 elemento = A(2,3);           % elemento da 2ª linha, 3ª coluna
4
5 % Acesso a linhas ou colunas inteiras
6 linha2 = A(2,:);             % segunda linha
7 coluna3 = A(:,3);            % terceira coluna
```

Código 7: Exemplo de acesso a elementos de uma matriz.

Somar um valor a todos os elementos da matriz

É possível adicionar um número a todos os elementos de uma matriz de forma simples:

$$B = A + 10$$

```
1 matrizA = [1 2 3; 4 5 6]
2 matrizB = matrizA + 10;
3
4 %resultante
5 [11, 12, 13; 14, 15, 16]
```

Código 8: Somando 10 a todos os elementos da matriz.

Atenção!

Operação pode ser realizada com quaisquer um dos operadores aritméticos.

Multiplicação de Matrizes vs. Elemento a Elemento I

› Multiplicação matricial ($A * B$):

›› Segue as regras da álgebra linear.

›› O número de colunas de A deve ser igual ao número de linhas de B.

```
1 A = [1 2 3; 4 5 6]; % 2x3
2 B = [2 2; 3 3; 4 4]; % 3x2
3 C = A * B;
4 % Saída esperada:
5 % Resultado C = A * B:
6 %      20      20
7 %      47      47
```

Código 9: Exemplo de multiplicação de matrizes.

Multiplicação de Matrizes vs. Elemento a Elemento II

- › Multiplicação elemento a elemento ($A \text{ .* } B$):
 - » Cada elemento de A é multiplicado pelo elemento correspondente de B.
 - » Matrizes devem ter o mesmo tamanho.
 - » Para divisão elemento a elemento, use $A ./ B$
 - » Para potência elemento a elemento, use $A.^B$

Multiplicação de Matrizes vs. Elemento a Elemento III

```
1 A = [1 2 3; 4 5 6];  
2 B = [2 2 2; 3 3 3];  
3 C = A .* B;  
4 % Saída esperada:  
5 % Resultado C = A .* B:  
6 %      2      4      6  
7 %     12     15     18
```

Código 10: Exemplo de multiplicação de elemento por elemento.

Funções Matriciais

A **transposta** de uma matriz troca linhas por colunas. No MATLAB, usa-se o operador `'`.

```
1 matriz = [1 2 3; 4 5 6]
2 matriz_transposta = matriz'
3 % Matriz resultante: [1 4; 2 5; 3 6]
```

Código 11: Exemplo de transposta de uma matriz.

A **inversa** de uma matriz quadrada A é a matriz A^{-1} que satisfaz:

$$A \cdot A^{-1} = I$$

```
1 matriz = [1 2 3; 0 1 4; 5 6 0];  
2 matriz_inversa = inv(A);  
3 % Saída esperada:  
4 % Inversa de A:  
5 %      -24      18      5  
6 %      20      -15     -4  
7 %      -5       4       1
```

Código 12: Exemplo de inversa de uma matriz.

Comando	Descrição
<code>eig(A)</code>	Calcula autovalores e autovetores
<code>inv(A)</code>	Calcula a inversa de A
<code>expm(A)</code>	Matriz exponencial e^A
<code>det(A)</code>	Determinante da matriz
<code>size(A)</code>	Tamanho da matriz
<code>lu(A)</code>	Fatorização triangular LU
<code>sqrtm(A)</code>	Raiz quadrada da matriz
<code>poly(A)</code>	Polinômio característico de A
<code>diag(A)</code>	Elementos da diagonal principal
<code>[V,D] = eig(A)</code>	V: autovetores, D: autovalores

Tabela 1: Principais funções para operações com matrizes.

(Chapman, 2016)



Bibliografia



CHAPMAN, Stephen J. **Programação em MATLAB para Engenheiros**. 5. ed. São Paulo: Thomson, 2016.



MORAIS, V.; VIEIRA, C. **MATLAB Curso Completo**. [S. l.]: FCA, 2013. 644 p.

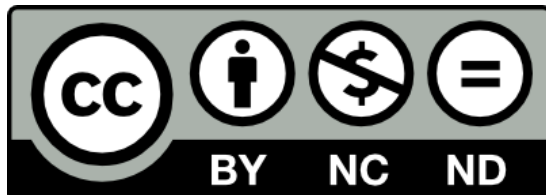


PALM, William J. **Introdução ao MATLAB para engenheiros**. 3. ed. Porto Alegre: AMGH, 2013. Tradução de Tales Argolo Jesus. xiv, 562 p.



MATHWORKS, INC. **MATLAB Help Center**. [*S. l.: s. n.*], 2025.
<https://www.mathworks.com/help/index.html>. Acesso em: 14 ago. 2025.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Programação para Engenharia II

Estruturas Homogêneas