

Programação para Engenharia II

Função

Sumário

- 1 Função
- 2 Função Nomeada
- 3 Função com Retorno

- 4 Função Anônima
- 5 Função Recursiva
- 6 Bibliografia

Função

Funções são blocos de código que executam uma tarefa específica e podem ser reutilizadas diversas vezes.

- Uma **função** recebe **valores de entrada**, realiza uma **operação** e pode retornar um **resultado**.
- São usadas para **organizar o código**, evitar repetições e facilitar a leitura do programa.
- O MATLAB possui diversas **funções prontas** (nativas) que podemos usar diretamente.

Exemplos de funções que já utilizamos:

```
1  vetor = [5, 10, 3, 8, 2];  
2  menor = min(vetor)      % Retorna o menor valor do vetor  
3  maior = max(vetor)      % Retorna o maior valor do vetor  
4  somaTotal = sum(vetor)  % Soma todos os elementos  
5  media = mean(vetor)    % Calcula a média aritmética
```

Saídas esperadas:

```
menor = 2  
maior = 10  
somaTotal = 28  
media = 5.6
```

Função Nomeada

Uma função nomeada é definida em um arquivo separado com a palavra-chave `function`.

- › O nome do arquivo deve ser igual ao nome da função.
- › Possui seu próprio espaço de trabalho.
- › Pode ou não retornar valores.

```
1 %Função Nomeada
2 function meu_max(vetor)
3     maximo = vetor(1);
4     for i = 2:length(vetor)
5         if vetor(i) > maximo
6             maximo = vetor(i);
7         end
8     end
9     disp(maximo);
10 end
```

Código 1: Função nomeada que encontra o maior valor de um vetor.

Exemplo de chamada no script principal:

```
1 %Chamando a função Nomeada
2 meuVetor = randi(50,1,20);
3
4 meu_max(meuVetor);
```

Código 2: Script que cria um vetor aleatório e chama a função meu_max.

Função com Retorno

Funções nomeadas podem retornar valores para o programa principal.

```
1 %Função nomeada com retorno.  
2 function retorno = multiplicar(a, b)  
3     resultado = a * b;  
4     retorno = resultado;  
5 end
```

Código 3: Função que retorna o produto entre dois números.

```
1 %Chamando função nomeada e armazenando seu retorno.  
2 multi = multiplicar(2,3);  
3 disp(multi)
```

Código 4: Script que chama a função multiplicar e exibe o resultado.

Função Anônima

Uma **função anônima** é uma função simples e de uma única linha, definida diretamente no código, sem precisar criar um arquivo separado (.m).

- É criada com o operador @.
- Ideal para operações matemáticas curtas e rápidas.
- Pode receber argumentos e retornar um valor.

Sintaxe:

```
1 nome_funcao = @(argumentos) expressao
```

Exemplo:

```
1 quadrado = @(x) x.^2; % Define a função
2 resultado = quadrado(5) % Chama a função
```

Saída:

```
resultado = 25
```

```
1 %Função Anônima.  
2 soma = @(a,b) a + b  
3  
4 soma(10,13)  
5 soma(2,4)
```

Código 5: Exemplo de função anônima que soma dois números.


```
1 %Função Anônima.  
2 formula = @(a,b,c) ((a + b) *(sqrt(25)) + (38^3)) / (1/7) * c;  
3  
4 formula(10,12,38)  
5 formula(2,4,15)
```

Código 6: Função anônima com expressão matemática mais complexa.

Função Nomeada

- › Definida em um arquivo `.m`.
- › Usa a palavra-chave `function`.
- › Pode ter várias linhas de código.
- › Suporta estruturas de controle (`if`, `for`, etc.).
- › Pode ter entradas e saídas múltiplas.

Função Anônima

- › Definida dentro do script principal.
- › Não precisa de arquivo separado.
- › Limitada a uma única expressão.
- › Ideal para operações rápidas e simples.

Quando usar cada uma:

- **Função Nomeada:** quando o código é longo, será reutilizado ou precisa de mais de uma linha.
- **Função Anônima:** geralmente para expressões ou formulas.

Função Recursiva

Uma função iterativa utiliza laços para repetir cálculos.

```
1 %Função Iterativa
2 function retorno = potencia(base, expoente)
3     potencia = 1;
4     for(i = (1:expoente))
5         potencia = potencia * base;
6     end
7     retorno = potencia;
8 end
```

Código 7: Função iterativa que calcula a potência de um número.

```
1 %Chamada Função Nomeada com retorno.  
2 pot = potencia(4,3);  
3 disp(pot);
```

Código 8: Script que chama a função iterativa potencia.

- A função deve chamar a si mesma;
- Deve possuir um ou mais casos-base, que são as condições que interrompem a recursão;
- Cada chamada recursiva deve aproximar-se do caso-base, modificando os parâmetros;
- O problema deve ser divisível em subproblemas menores semelhantes ao original;
- O resultado final é obtido a partir da combinação dos resultados das chamadas recursivas;
- A função deve garantir que não ocorra recursão infinita, ou seja, que o caso-base seja sempre alcançado.

Uma função recursiva chama a si mesma para resolver subproblemas menores.

```
1 function resultado = potencia_recursiva(base, expoente)
2     if expoente == 0
3         resultado = 1;
4     else
5         resultado = base * potencia_recursiva(base, expoente -
6             ↪ 1);
7     end
8 end
```

Código 9: Função recursiva que calcula a potência de um número.

(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.

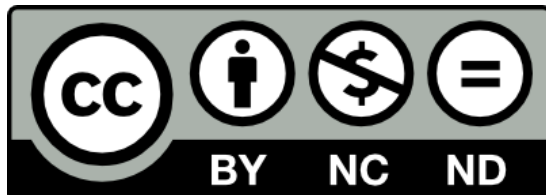


PALM, William J. **Introdução ao MATLAB para engenheiros**. 3. ed. Porto Alegre: AMGH, 2013. Tradução de Tales Argolo Jesus. xiv, 562 p.



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

Função