

INTRODUÇÃO À PROGRAMAÇÃO

Aula 5

Passos na resolução de problemas de engenharia

1. Entenda o objetivo do problema.
2. Liste as informações conhecidas.
3. Determine quais informações você precisa encontrar.
4. Simplifique o problema apenas o suficiente para obter a informação necessária. (Informações desprezíveis, Propriedades físicas constantes...)
5. Desenhe um esboço e nomeie todas as variáveis necessárias.
6. Determine quais princípios fundamentais são aplicáveis. (Equações)
7. Pense de uma maneira geral sobre a abordagem de solução do problema e considere outras abordagens antes de prosseguir. (Equações adequadas)

Passos na resolução de problemas de engenharia

8. Dê o nome a cada passo no processo de resolução.
9. Se você resolver o problema com um programa, se possível verifique a mão os resultados utilizando uma versão simples do problema.
10. Verifique se sua resposta faz sentido. Estime um faixa de resultados esperados e compare com sua resposta.

Arranjos numéricos

- Um dos pontos fortes do MATLAB é sua capacidade de manipular coleções de itens, chamados de arranjos, como se elas fossem uma entidade única.
- A funcionalidade de manipulação de arranjos significa que os programas em MATLAB podem ser muito pequenos.
- O Arranjo é o bloco de construção básico no MATLAB.

Vetores no MATLAB

- No MATLAB, um vetor é representado por um arranjo unidimensional.

`[5, 4, 2]` ou `[5 4 2]` -> vetor linha

`[5;4;2]` ou `[5, 4, 2]'` -> vetor coluna

- Para criar um vetor no MATLAB é necessário o uso de colchetes, exceto no caso em que for utilizado o operador `(:)`.

Vetores no MATLAB

- No MATLAB um vetor é apenas uma lista de escalares.
- Outra forma de criar um vetor coluna:

```
>> g = [3,  
4,  
5,  
6]
```

```
g =
```

```
3  
4  
5  
6
```

Vetores no MATLAB

- É possível criar vetores anexando um vetor ao outro.

```
>> r = [2, 4, 20]
```

```
r =
```

```
2    4   20
```

```
>> w = [3 , 6, 9]
```

```
w =
```

```
3    6    9
```

```
>> u = [r,w]
```

```
u =
```

```
2    4   20    3    6    9
```

Vetores no MATLAB

- O operador dois pontos (:) gera um vetor de elementos regularmente espaçados.

`x = m:q:n`

- Um vetor **x** de valores com espaçamento **q** é criado.
- O primeiro valor é **m**.
- O último valor é **n** se $|m - n|$ for um múltiplo inteiro de **q**. Se não, o último valor é menor do que **n**.

`>> x = 0:2:8`

`x =`

0 2 4 6 8

Vetores no MATLAB

`x = m:q:n`

- Um vetor **x** de valores com espaçamento **q** é criado.
- O primeiro valor é **m**.
- O último valor é **n** se $|m - n|$ for um múltiplo inteiro de **q**. Se não, o último valor é menor do que **n**.

`>> x = 0:2:7`

`x =`

0 2 4 6

Vetores no MATLAB

`x = m:q:n`

Quando o incremento q é omitido, presume-se que ele é igual a 1.

```
>> y = 5:10
```

y =

5 6 7 8 9 10

```
>> y = -3:2
```

y =

-3 -2 -1 0 1 2

Vetores no MATLAB

`x = m:q:n`

O incremento **q** pode ser negativo. Nesse caso, **m** deve ser maior que **n**.

```
>> u = 10:-2:4
```

u =

10 8 6 4

```
>> p = 8:-1:0
```

p =

8 7 6 5 4 3 2 1 0

Vetores no MATLAB

- O comando ***linspace*** também cria um vetor linha com espaçamento linear.
- Em vez de especificar o incremento, agora é especificado o número de valores.

`linspace (x1, x2, n)`

- **x1** e **x2** são os limites inferior e superior
- **n** é o número de pontos
- Ex: `linspace(5, 6, 11)` é o mesmo que `5:0.1:6`

```
>> linspace (5, 6, 11)
```

```
ans =
```

```
Columns 1 through 7
```

```
5.0000 5.1000 5.2000 5.3000 5.4000 5.5000 5.6000
```

```
Columns 8 through 11
```

```
5.7000 5.8000 5.9000 6.0000
```

Arranjos Bidimensionais

- Um arranjo que tenha linhas e colunas é um arranjo bidimensional, que na maioria das vezes é também uma matriz.

- $$A = \begin{bmatrix} 1 & 5 \\ 9 & -1 \\ 2 & 4 \end{bmatrix}$$

- Tamanho do arranjo é número de linhas e o número de colunas.
- Por exemplo um arranjo de 3 linhas e 2 colunas é chamado de arranjo 3 x 2
- Um elemento da matriz A pode ser representado por a_{ij} , i é o número da linha e j é o número da coluna.
- Por exemplo o número a_{32} está na linha 3 e na coluna 2.

Criando Matrizes

```
>> A = [2, 3, 10; 3, 7, 12]
```

```
A =
```

```
2  3 10  
3  7 12
```

```
>> A = [2, 3, 10  
3, 7, 12]
```

```
A =
```

```
2  3 10  
3  7 12
```

Criando Matrizes

```
>> a = [1, 3, 5]
```

```
a =
```

```
1    3    5
```

```
>> b = [7, 9, 11]
```

```
b =
```

```
7    9   11
```

```
>> c = [a, b]
```

```
c =
```

```
1    3    5    7    9   11
```

```
>> D = [a; b]
```

```
D =
```

```
1    3    5  
7    9   11
```

Matrizes e operação de transposição

```
>> A = [2, 3, 10; 3, 7, 12]
```

```
A =
```

```
     2     3    10  
     3     7    12
```

```
>> A'
```

```
ans =
```

```
     2     3  
     3     7  
    10    12
```


Endereçamento de arranjos

Os índices de um arranjo são os números da linha e da coluna de um elemento em um arranjo, e são utilizados para localizar os seus elementos.

```
>> v = [3, 7, 5, 10]
```

```
v =
```

```
    3    7    5   10
```

```
>> v(2)
```

```
ans =
```

```
    7
```

```
>> v(4)
```

```
ans =
```

```
   10
```

```
>> v(8)
```

```
Index exceeds matrix dimensions.
```

Endereçamento de arranjos

`v(:)` - representa todos os elementos da linha ou da coluna do vetor `v`

```
>> v = [3, 7, 5, 10];
```

```
>> v(:)
```

```
ans =
```

```
3
```

```
7
```

```
5
```

```
10
```

```
>> v(1:3)
```

```
ans =
```

```
3
```

```
7
```

```
5
```

Endereçamento de arranjos

```
>> A = [2, 3, 5;4, 7, 10;7, 8, 20]
```

```
A =
```

2	3	5
4	7	10
7	8	20

```
>> A(2,1)
```

```
ans =
```

4

```
>> A(3,3)
```

```
ans =
```

20

Endereçamento de arranjos

```
>> A = [2, 3, 5;4, 7, 10;7, 8, 20]
```

```
A =
```

2	3	5
4	7	10
7	8	20

```
>> A(:,2)
```

```
ans =
```

3
7
8

```
>> A(3,:)
```

```
ans =
```

7	8	20
---	---	----

Endereçamento de arranjos

```
>> A = [2, 3, 5;4, 7, 10;7, 8, 20]
```

```
A =
```

2	3	5
4	7	10
7	8	20

```
>> A(:,2:3)
```

```
ans =
```

3	5
7	10
8	20

Endereçamento de arranjos

```
>> A = [2, 3, 5;4, 7, 10;7, 8, 20]
```

```
A =
```

2	3	5
4	7	10
7	8	20

```
>> A(:)
```

```
ans =
```

2
4
7
3
7
8
5
10
20

Endereçamento de arranjos

```
>> A = [2, 3, 5;4, 7, 10;7, 8, 20]
```

```
A =
```

2	3	5
4	7	10
7	8	20

```
>> A(end, :)
```

```
ans =
```

7	8	20
---	---	----

```
>> A(:,end)
```

```
ans =
```

5
10
20

Endereçamento de arranjos

```
>> B = [2, 4, 10, 13;16, 3, 7, 18; 8, 4, 9, 25; 3, 12, 15, 17]
```

B =

2	4	10	13
16	3	7	18
8	4	9	25
3	12	15	17

```
>> C = B(2:3,1:3)
```

C =

16	3	7
8	4	9