

Computação Aplicada à Engenharia

Prof. Claudio Teodoro dos Santos

Arranjos (1)

Tópicos

- Criação de arranjo de números:
 - Vetores: arranjos 1-D
 - Matrizes: arranjos 2-D
- Endereçamento de arranjos
- Strings e Variáveis de String

Nota

- No MATLAB todas as variáveis são armazenadas como arranjos.
- Se o valor de uma variável é um único número, um arranjo $|x|$ é usado.

Arranjos de números são usados em muitas aplicações.

Exemplos:

Um arranjo de números pode representar dados:

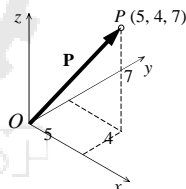
Ano	1984	1986	1988	1990	1992	1994	1996
População	127	130	136	145	158	178	211

Um arranjo de números pode representar um vetor.

Um exemplo é o vetor de posição. A localização do ponto P em um espaço tridimensional pode ser representada por três coordenadas cartesianas, 5, 4 e 7.

O vetor de posição, que aponta para a localização do ponto P em relação ao ponto O (origem do sistema de coordenadas), é definido por:

$$\mathbf{P} = 5\mathbf{i} + 4\mathbf{j} + 7\mathbf{k}$$



3

No MATLAB, um vetor (ou qualquer lista de números) pode ser definido como um vetor linha (horizontal) ou coluna (vertical).

Por exemplo, os dados de população do slide anterior podem ser definidos como vetores linha:

```
[1984 1986 1988 1990 1992 1994 1996]
```

```
[127 130 136 145 158 178 211]
```

ou como vetores coluna:

```
[1984
1986
1988
1990
1992
1994
1996]
```

```
[127
130
136
145
158
178
211]
```

O vetor de posição pode ser definido como:

linha: [5 4 7]

coluna: $\begin{bmatrix} 5 \\ 4 \\ 7 \end{bmatrix}$

4

criação de vetor no MATLAB

Um vetor é criado ao se digitar os elementos (números) dentro do espaço entre colchetes [].

Para criar um **vetor linha** digite um espaço ou uma vírgula entre os elementos dentro dos colchetes.

```
>> yr=[1984 1986 1988 1990 1992 1994 1996] ← Tecla e pressione Enter
yr =
    1984    1986    1988    1990    1992    1994    1996 ← Resposta do computador
```

```
>> cor = [5,4,7] ← Tecla e pressione Enter
cor =
     5     4     7 ← Resposta do computador
```

O MATLAB não é "exigente" em relação ao modo como os dados são digitados. Pode-se usar espaços antes e/ou depois do sinal =. Entre os elementos, pode-se usar um ou mais espaços após a vírgula.

5

Para criar um **vetor coluna** digite o colchete [e então digite os elementos com um ponto-e-vírgula entre eles, ou pressione **Enter** após cada elemento. Digite o colchete] após o último elemento.

```
>> pop = [127; 130; 136; 145; 158; 178; 211] ← Tecla e pressione Enter
pop =
    127
    130
    136
    145
    158
    178
    211 ← Resposta do computador

>> cor = [5
4
7] ← Tecla e pressione Enter após o 5, o 4 e depois o ].
cor =
     5
     4
     7 ← Resposta do computador
```

6

CRIAÇÃO DE UM VETOR DE PASSO CONSTANTE

Em um vetor de passo constante, a diferença entre os elementos é sempre a mesma, (p. ex., $v = 2\ 4\ 6\ 8\ 10\ 12$).

Um vetor em que o primeiro termo é m , o passo é q e o último termo é n pode ser criado ao se digitar $[m:q:n]$.

```
>> x = [1:2:13]
x =
    1    3    5    7    9   11   13
```

```
>> x = [1.5:0.1:2.1]
x =
 1.5000 1.6000 1.7000 1.8000 1.9000 2.0000 2.1000
```

Se o passo é omitido o padrão (default) é 1.

```
>> x = [-3:7]
x =
   -3   -2   -1    0    1    2    3    4    5    6    7
```

7

CRIANDO UM VETOR PELA ESPECIFICAÇÃO DO PRIMEIRO E ÚLTIMO TERMOS E DO NÚMERO DE TERMOS

Um vetor em que o primeiro termo é xi , o último termo é xf e o número de termos igualmente espaçados é n , pode ser criado ao se digitar $linspace(xi,xf,n)$.

```
>> u = linspace(0,8,6)
u =
    0  1.6000  3.2000  4.8000  6.4000  8.0000
```

Se o número de termos é omitido, o default é 100.

Digite:

```
>> u = linspace(0,49,5)
```

pressione **Enter** e veja a resposta do computador.

Ela deve ser:

```
u = 0  0.5000  1.0000  1.5000 ... (100 termos) ... 49.0000 49.5000
```

8

ARRANJO BIDIMENSIONAL - MATRIZ

Uma matriz é um arranjo bidimensional de números.

Em uma matriz quadrada, o número de linhas e colunas é igual:

```
7 4 9
3 8 1
6 5 3
```

Três linhas e três colunas (3x3)

Geralmente, o número de linhas e colunas é diferente:

```
31 26 14 18 5 30
 3 51 20 11 43 65
28  6 15 61 34 22
14 58  6 36 93  7
```

Quatro linhas e seis colunas (4x6)

$(m \times n)$ matriz tem m linhas e n colunas

$(m \times n)$ é chamado de tamanho (size) da matriz

9

CRIAÇÃO DE MATRIZ NO MATLAB

Uma matriz é criada ao se digitar os elementos (números) linha por linha dentro de colchetes [].

Tecla o colchete [, e então digite a primeira linha separando os elementos com espaço ou vírgula. Para digitar a próxima linha, digite um ponto-e-vírgula ou pressione **Enter**. Tecla o colchete] no fim da última linha.

```
>> a=[1 2 3; 4 5 6; 7 8 9]
a =
    1    2    3
    4    5    6
    7    8    9
```

Tecla e pressione **Enter**
Resposta do computador

```
>> b=[11 12 13 14 15
16 17 18 19 20
21 22 23 24 25]
b =
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25
```

Tecla e pressione **Enter** após cada linha e após o].
Resposta do computador

10

A OPERAÇÃO DE TRANSPOSIÇÃO

A operação de transposição ' .

Para um vetor: Converte vetor linha em vetor coluna, ou vice versa.

Para uma matriz: Troca linhas por colunas.

Exemplo para um vetor:

```
>> a = [3 8 1]
a =
     3     8     1
>> b = a'
b =
     3
     8
     1
```

11

A OPERAÇÃO DE TRANSPOSIÇÃO

Exemplo para uma matriz:

```
>> c = [2 55 14 8; 21 5 32 11; 41 64 9 1]
c =
     2    55    14     8
    21     5    32    11
    41    64     9     1
>> d = c'
d =
     2    21    41
    55     5    64
    14    32     9
     8    11     1
```

12

ENDEREÇAMENTO DE ARRANJO (VETOR)

O endereço de um elemento em um vetor é sua posição na linha (ou coluna).
Para o vetor "v", v(k) refere-se ao elemento na posição k. A primeira posição é 1.

```
>> v = [35 46 78 23 5 14 81 3 55]
v =
    35    46    78    23     5    14    81     3    55
```

```
>> v(4)
ans =
    23
```

```
>> v(7)
ans =
    81
```

```
>> v(1)
ans =
    35
```

É possível alterar um elemento de um vetor definindo um valor para um endereço específico diretamente:

```
>> v(6)=273
v =
    35    46    78    23     5   273    81     3    55
```

Elementos podem ser usados como variáveis em cálculos:

```
>> v(2)+v(8)
ans =
    49
```

```
>> v(5)^v(8)
ans =
   125
```

13

ENDEREÇAMENTO DE ARRANJO (MATRIX)

O endereço de um elemento em uma matriz é sua posição, definida pelo número da linha e pelo número da coluna.

Para uma matriz "m", m(k,p) refere-se ao elemento na linha k e coluna p.

```
>> m=[3 11 6 5; 4 7 10 2; 13 9 0 8]
m =
     3    11     6     5
     4     7    10     2
    13     9     0     8
```

```
>> m(1,1)
ans =
     3
```

```
>> m(2,3)
ans =
    10
```

É possível alterar um elemento de uma matriz definindo um valor para um endereço específico diretamente:

```
>> m(3,1)=20
m =
    20    11     6     5
     4     7    10     2
    13     9     0     8
```

Elementos podem ser usados como variáveis em cálculos:

```
>> m(2,4)-m(1,2)
ans =
    -9
```

14

USO DE (:) NO ENDEREÇAMENTO DE ARRANJOS

O ":" pode ser usado para endereçar uma faixa de elementos em um vetor ou matriz.

Para um vetor:

v(:) Representa todos os elementos do vetor (seja vetor linha ou vetor coluna)

v(3:6) Representa os elementos 3 até 6. (i.e., v(3), v(4), v(5), v(6)).

```
>> v = [4 15 8 12 34 2 50 23 11]
v =
     4    15     8    12    34     2    50    23    11
>> u = v(3:7)
u =
     8    12    34     2    50
```

15

USO DE (:) NO ENDEREÇAMENTO DE ARRANJOS

Para uma matriz:

A(:, 3) Refere-se aos elementos de todas as linhas da coluna 3.

A(2, :) Refere-se aos elementos de todas as colunas da linha 2.

A(:, 2:5) Refere-se aos elementos de todas as linhas das colunas 2 até 5.

A(2:4, :) Refere-se aos elementos das linhas 2 até 4 de todas as colunas.

A(1:3, 2:4) Refere-se aos elementos das linhas 1 até 3 das colunas 2 até 4.

16

EXEMPLOS DO USO DE (:) NO ENDEREÇAMENTO DE ARRANJOS

Define uma matriz

```
>> A = [1 3 5 7 9; 2 4 6 8 10;
        3 6 9 12 15; 4 8 12 16 20;
        5 10 15 20 25]
A =
     1     3     5     7     9
     2     4     6     8    10
     3     6     9    12    15
     4     8    12    16    20
     5    10    15    20    25
```

```
>> B = A(:,3)
B =
     5
     6
     9
    12
    15
```

```
>> C = A(2,:)
C =
     2     4     6     8    10
```

17

EXEMPLOS DO USO DE (:) NO ENDEREÇAMENTO DE ARRANJOS (CONT.)

```
A =
     1     3     5     7     9
     2     4     6     8    10
     3     6     9    12    15
     4     8    12    16    20
     5    10    15    20    25
```

```
>> E = A(2:4,:)
E =
     2     4     6     8    10
     3     6     9    12    15
     4     8    12    16    20
```

```
>> D = A(:, 2:5)
D =
     3     5     7     9
     4     6     8    10
     6     9    12    15
     8    12    16    20
    10    15    20    25
```

```
>> F = A(1:3,2:4)
F =
     3     5     7
     4     6     8
     6     9    12
```

18

ALGUMAS NOTAS IMPORTANTES SOBRE VARIÁVEIS

- Todas as variáveis no MATLAB são arranjos. Um **escalar** é um arranjo com um elemento, um **vetor** é um arranjo com uma linha ou uma coluna de elementos, e uma **matriz** é um arranjo de linhas e colunas de elementos.
- O tipo de variável é definido na entrada, quando a variável é criada (tipagem dinâmica).
- O elemento (escalar) ou os elementos (vetor, matriz) de uma variável pode(m) conter números (reais, complexos) ou expressões.
- O comando "who" mostra quais variáveis estão armazenadas na memória.
- O comando "whos" lista as variáveis armazenadas, o tipo de cada variável e a quantidade de memória utilizada por elas.

19

EXEMPLO

```
>> a = 7
a =
    7

>> E = 3
E =
    3

>> d = [5 a+E 4 E^2]
d =
    5    10    4    9

>> g = [a a^2 13; a*E 1 a^E]
g =

    7    49    13
   21     1   343
```

```
>> who
```

Your variables are:

E a d g

```
>> whos
```

Name	Size	Bytes	Class
E	1x1	8	double array
a	1x1	8	double array
d	1x4	32	double array
g	2x3	48	double array

Grand total is 12 elements using 96 bytes

20

STRINGS E VARIÁVEIS DE STRING

- ❖ Strings são caracteres colocados entre apóstrofes.
- ❖ Uma string pode conter letras, números, símbolos e espaços.
- ❖ Exemplos de strings: 'ad ef', '3%fr2', '{edcba :21!'.
- ❖ Strings podem ser usadas para atribuir variáveis.
- ❖ As strings são usadas como entrada de algumas funções.

21

VARIÁVEIS DE STRING (OU DE TEXTO)

Uma variável pode ser definida como uma string ao se digitar:

Nome da variável = 'texto'

```
>> a = 'ERty 8'
a =
ERty 8
```

```
>> B = ['Meu nome eh Joao Silva']
B =
Meu nome eh Joao Silva
```

- As strings são armazenadas como vetores linha nos quais todo caracter (incluindo espaços) é um elemento.
- Nas variáveis acima, *a* tem 6 elementos e *B* tem 22 elementos.
- Os elementos podem ser endereçados diretamente, tal como nos vetores numéricos.
- Nas variáveis acima:

```
>> a(4)
ans =
y
```

```
>> B(13)
ans =
J
```

22

VARIÁVEIS DE STRING

A variável de string:

```
>> x = '536'
x =
536
```

Não é a mesma coisa que a variável de número:

```
>> x = 536
x =
536
```

A variável numérica pode ser usada em cálculos enquanto que a variável de string (tipo caracter) não pode.

Uma importante aplicação das strings é a criação de mensagens de entrada e saída e dados.

23