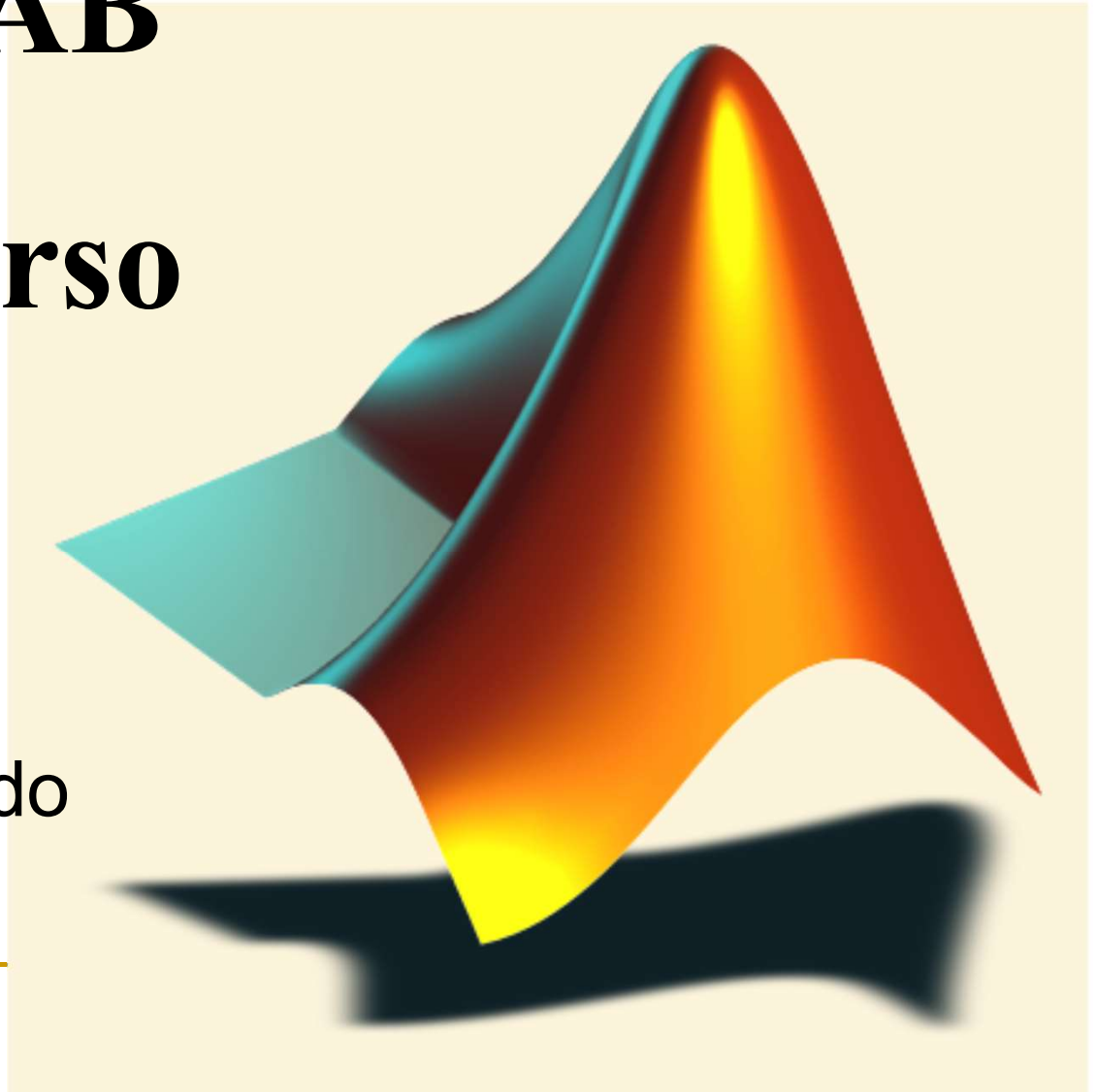

MATLAB

Mini-curso

Paulo Salgado



MatLab

O MATLAB é um ambiente de programação de alto nível para aplicações Científicas e de Engenharia.

Facilidades

- ✓ Oferece um leque alargado de bibliotecas de funções pré-definidas.
- ✓ Muito amigável em funcionalidades gráficas para Visualização de Dados.
- ✓ Largamente divulgado em Universidades e Laboratórios de Investigação.
- ✓ Muito conveniente para o desenvolvimento eficaz de protótipos.

O que é o MATLAB?

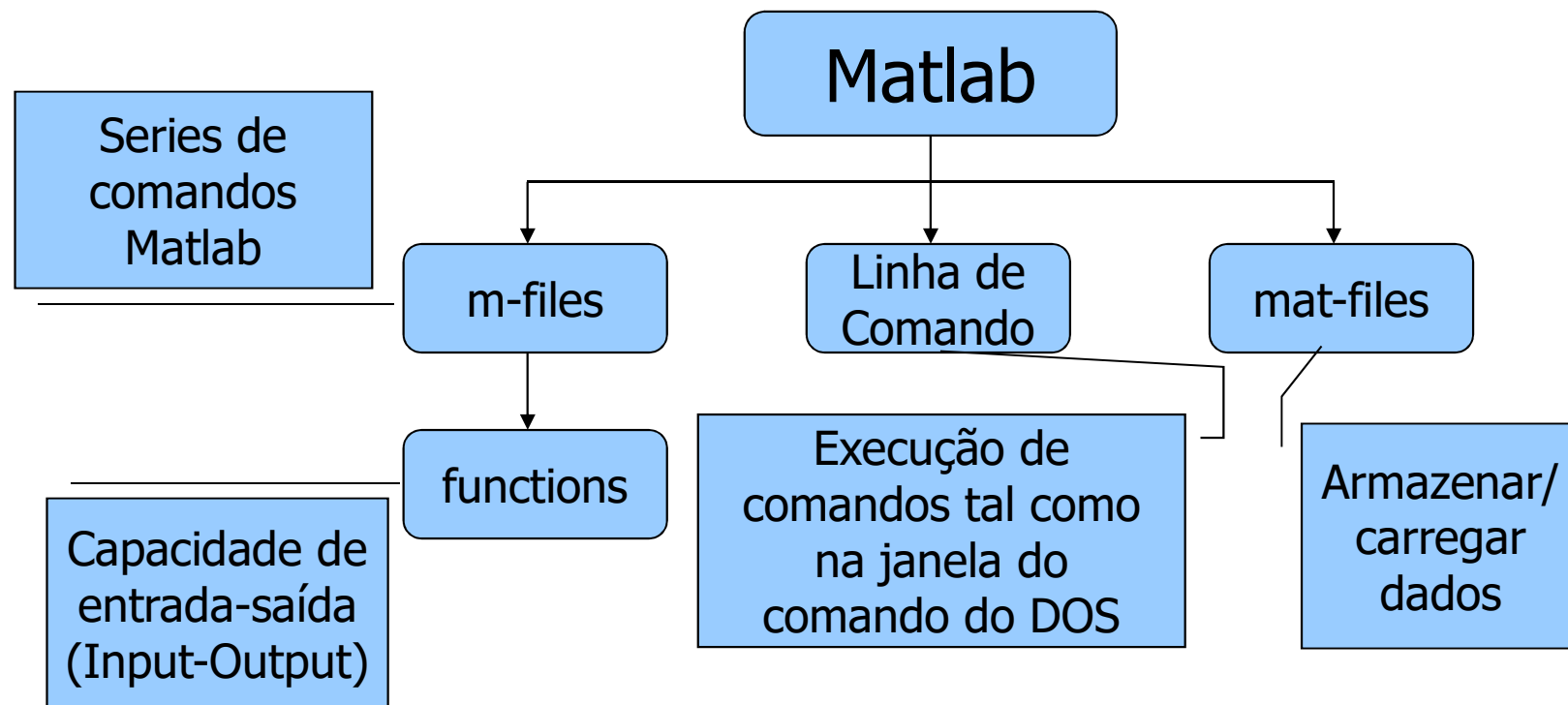
Utilidade

- Software de alto-desempenho
 - **Computação**
 - **Visualização**
 - **Ambiente de trabalho.**
- Linguagem de alto-nível

Suporta

- **Tipos dados**
- **Funções**
- **Instruções de Controlo fluxo**
- **Input/output**
- **Gráficos**
- **Programação Orientada a Objectos**

De que modo será utilizado?



Conteúdo:

- Janela do Matlab
- Variáveis, array, matrizes, índices
- Operadores (Aritmética, relacional, lógicos)
- Facilidades gráficas
- Controlo de fluxo
- Criar e utilizar as M-Files
- Escrita de Funções
- Conclusões

Partes do MATLAB

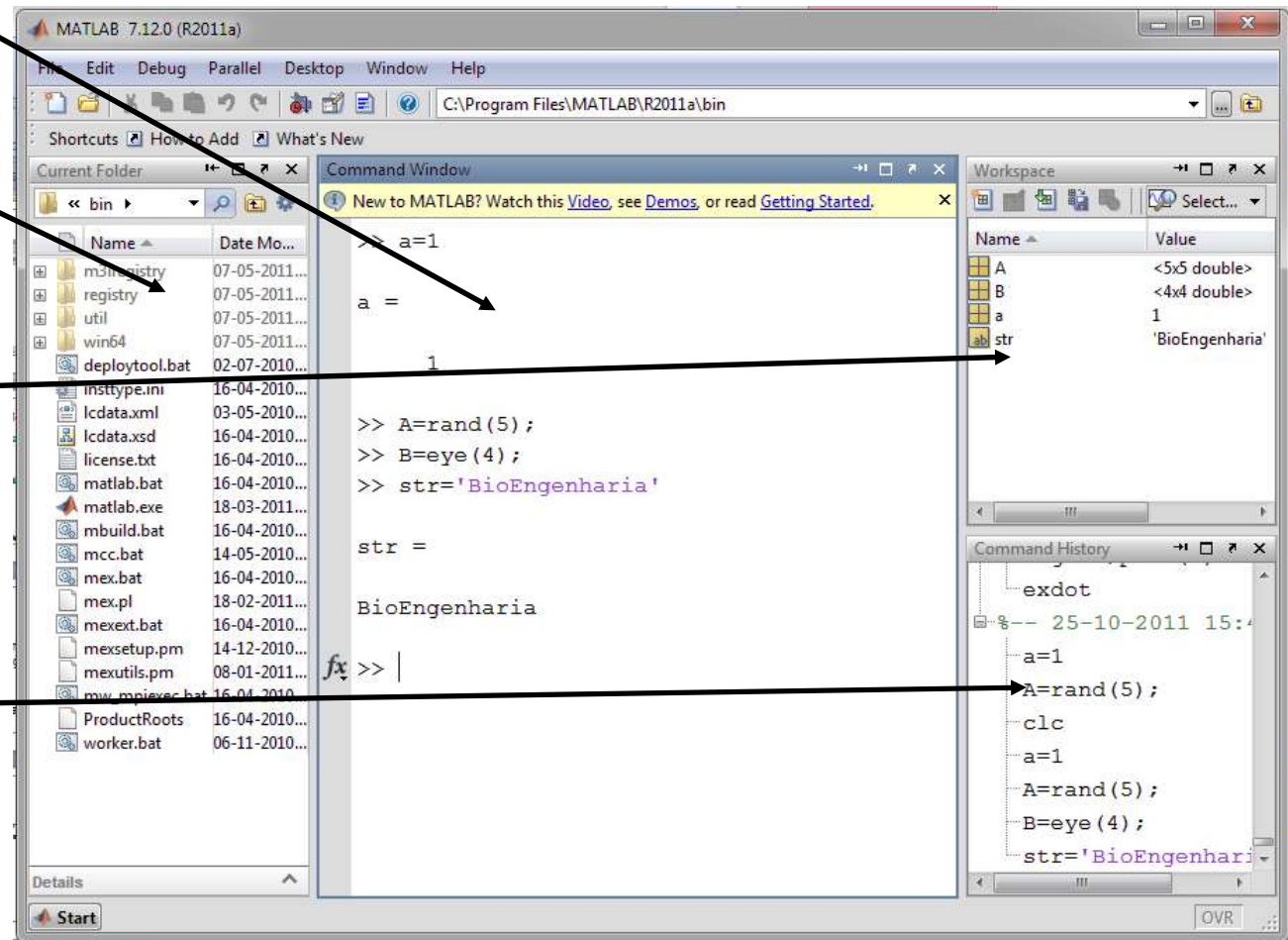
- Ambiente de Desenvolvimento
- Linguagem de Programação
- Gráficos
- Toolboxes
- Aplicações de Interface de Programas

Toolboxes

- Colecção de funções para resolver determinados problemas subordinado a aplicações específicas.
 - ❑ DSP Toolbox
 - ❑ Image Toolbox
 - ❑ Wavelet Toolbox
 - ❑ Neural Network Toolbox
 - ❑ Fuzzy Logic Toolbox
 - ❑ Control Toolbox
 - ❑ Communication Toolbox
 - ❑

Janela do Matlab

- Command Window
 - escrever comandos
- Current Folder (Directório actual)
 - Ver pastas, m-files e outros ficheiros
- Workspace (espaço de trabalho)
 - Ver as variáveis.
 - “Double click” na variável para as ver no “Array Editor”
- Command History
 - Ver comandos anteriores
 - Grava uma secção de trabalho



Calculos na Linha de Comandos

MATLAB como calculadora

```
>> -5/(4.8+5.32)^2
ans =
    -0.0488
>> (3+4i)*(3-4i)
ans =
    25
>> cos(pi/2)
ans =
    6.1230e-017
>> exp(acos(0.3))
ans =
    3.5470
```

Atribuição de Variables

```
>> a = 2;
>> b = 5;
>> a^b
ans =
    32
>> x = 5/2*pi;
>> y = sin(x)
y =
    1
>> z = asin(y)
z =
    1.5708
```

Ponto-virgula suspende a apresentação no ecrã

Resultado atribuído à variável "ans" se nenhum nome for especificado

() parênteses para funções com entradas

Funções Gerais

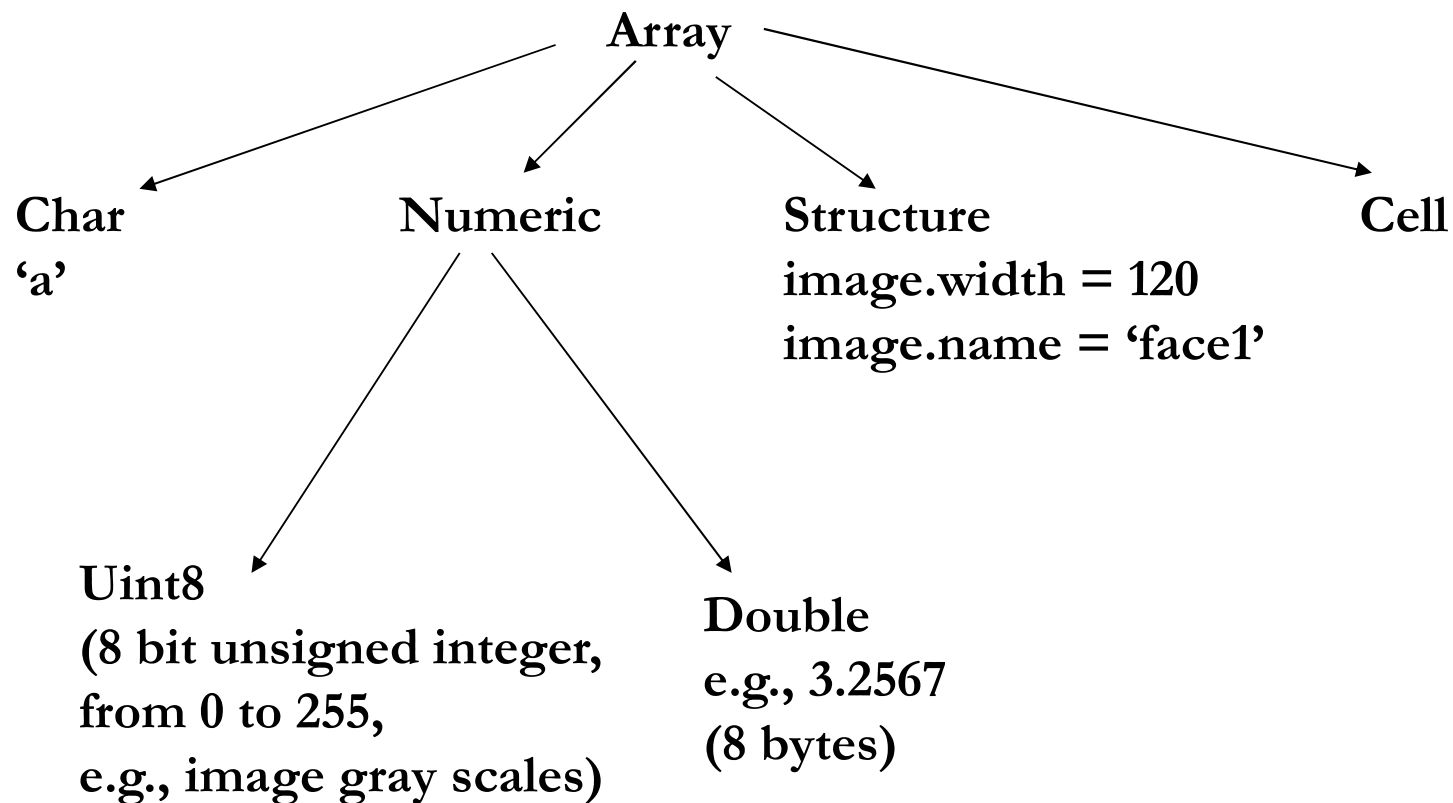
- **whos**: Lista as variáveis correntes
- **clear**: Limpa as variáveis e funções da memória
- **close**: Fecha a última figura
- **cd**: Muda de directoria de trabalho
- **dir**: Lista os ficheiros da directoria
- **echo**: Comando “Eco” nas M-files
- **format**: Formata o conjunto de saídas

Obter ajuda (help)

- *help* command (`>>help`)
- *lookfor* command (`>>lookfor`)
- Help Browser (`>>doc`)
- *helpwin* command (`>>helpwin`)
- Search Engine
- Printable Documents
 - “Matlabroot\help\pdf_doc\”
- Link to The MathWorks



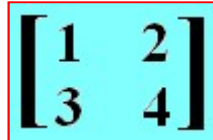
Tipos de Dados em MatLab



Matrizes

- Entrar e gerar Matrizes
- Subscripts
- Expansão escalar
- Concatenação
- Apagar linhas (“rows”) e colunas (“columns”)
- Extracção de “Array”
- Multiplicação de Matrizes e “Array”

Entrada de Matrizes Numéricas



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

Separador linha
Ponto e virgula (;)

Separador coluna
espaço / virgula (,)

```
>> a=[1 2;3 4]
a =
     1     2
     3     4

>> b=[-2.8, sqrt(-7), (3+5+6)*3/4]
b =
-2.8000    0 + 2.6458i    10.5000

>> b(2,5) = 23
b =
-2.8000    0 + 2.6458i    10.5000    0    0
         0              0          0    0   23.0000
```

Usar parentesis rectos []

- Qualquer expressão do MATLAB pode ter como elemento de entrada uma matriz.
- Matrizes devem ser rectangulares.

A Matriz no MATLAB

A =

| | | Columns (colunas) (n) | | | | |
|----------------------|---|--------------------------|------------------|------------------|-----------------|------------------|
| | | 1 | 2 | 3 | 4 | 5 |
| Rows (m) (linhas) | 1 | 4 ¹ | 10 ⁶ | 1 ¹¹ | 6 ¹⁶ | 2 ²¹ |
| | 2 | 8 ² | 1.2 ⁷ | 9 ¹² | 4 ¹⁷ | 25 ²² |
| | 3 | 7.2 ³ | 5 ⁸ | 7 ¹³ | 1 ¹⁸ | 11 ²³ |
| | 4 | 0 ⁴ | 0.5 ⁹ | 4 ¹⁴ | 5 ¹⁹ | 56 ²⁴ |
| | 5 | 23 ⁵ | 83 ¹⁰ | 13 ¹⁵ | 0 ²⁰ | 10 ²⁵ |

A (2,4)

A (17)

Matriz rectangular:
Escalar: 1-by-1 array
Vector: m-by-1 array
 1-by-n array
Matrix: m-by-n array

Entrada Numérica de Matrizes

Expansão escalar

» `w=[1 2;3 4] + 5`

`w =`

| | |
|---|---|
| 6 | 7 |
| 8 | 9 |

Gerar sequencia:

» `x = 1:5`

Operator 2 pontos (:)

`x =`

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

» `y = 2:-0.5:0`

`y =`

| | | | | |
|--------|--------|--------|--------|---|
| 2.0000 | 1.5000 | 1.0000 | 0.5000 | 0 |
|--------|--------|--------|--------|---|

**Funções uteis para
criar matrizes.**

» `z = rand(2,4)`

`z =`

| | | | |
|--------|--------|--------|--------|
| 0.9501 | 0.6068 | 0.8913 | 0.4565 |
| 0.2311 | 0.4860 | 0.7621 | 0.0185 |

Mais exemplos ...

- um vector $x = [1 \ 2 \ 5 \ 1]$

$$x = \begin{matrix} & 1 & 2 & 5 & 1 \end{matrix}$$

- uma matriz $x = [1 \ 2 \ 3; \ 5 \ 1 \ 4; \ 3 \ 2 \ -1]$

$$x = \begin{matrix} & 1 & 2 & 3 \\ 1 & 5 & 1 & 4 \\ 2 & 3 & 2 & -1 \end{matrix}$$

- transposta $y = x'$

$$y = \begin{matrix} 1 \\ 2 \\ 5 \\ 1 \end{matrix}$$

Mais exemples ...

■ `t = 1:10`

t =
1 2 3 4 5 6 7 8 9 10

■ `k = 2:-0.5:-1`

k =
2 1.5 1 0.5 0 -0.5 -1

■ `B = [1:4; 5:8]`

x =
1 2 3 4
5 6 7 8

Carregar/gravar dados

Ficheiros de dados

- **save** *filename var1 var2 ...*

>> save myfile.mat x y → binary

>> save myfile.dat x -ascii → ascii

- **load** *filename*

>> load myfile.mat → binary

>> load myfile.dat -ascii → ascii

- **Ficheiros Excel: xlsread e xlswrite**

>> [num,txt,row] = xlsread(filename,sheet,range)

>> xlswrite(filename,A,sheet,range)

Criar matrizes com o uso de funções

- `zeros(M,N)` MxN matrix of zeros

```
x = zeros(1,3)
```

```
x =
```

```
0      0      0
```

- `ones(M,N)` MxN matrix of ones

```
x = ones(1,3)
```

```
x =
```

```
1      1      1
```

- `rand(M,N)` MxN matrix of uniformly distributed random numbers on (0,1)

```
x = rand(1,3)
```

```
x =
```

```
0.9501  0.2311  0.6068
```

Dado:

```
A =  
  
     3     5     3  
     6     8     2  
     2     7     3
```

```
>> A(6)  
  
ans =  
  
     7
```

```
>> A(3,2)  
  
ans =  
  
     7
```

```
>> A(2,:)   
  
ans =  
  
     6     8     2
```

```
>> A(1:2,2)  
  
ans =  
  
     5  
     8
```

A(-2), A(0)

Error: ??? Subscript indices must either be real positive integers or logicals.

A(4,2)

Error: ??? Index exceeds matrix dimensions.

Concatenação numérica de Matrizes

Usar [] para combinar
matrizes existentes
como matrizes
“elementares”

Separador de linha:
Ponto e vírgula (;)

Separador de coluna:
espaço / virgula (,)

```
» a=[1 2;3 4]
a =
     1     2
     3     4
» cat_a=[a, 2*a; 3*a, 4*a; 5*a, 6*a]
cat_a =
     1     2     2     4
     3     4     6     8
     3     6     4     8
     9    12    12    16
     5    10     6    12
    15    20    18    24
```

Usar parêntesis rectos []

4*a

Nota: A matriz resultante terá que ser rectangular

Concatenação de Matrizes - Exemplos

■ $x = [1 \ 2], \ y = [4 \ 5], \ z = [0 \ 0]$

$$A = [x \ y]$$

$$\begin{matrix} 1 & 2 & 4 & 5 \end{matrix}$$

$$B = [x ; y]$$

$$\begin{matrix} 1 & 2 \\ 4 & 5 \end{matrix}$$

$$C = [x \ y ; z]$$

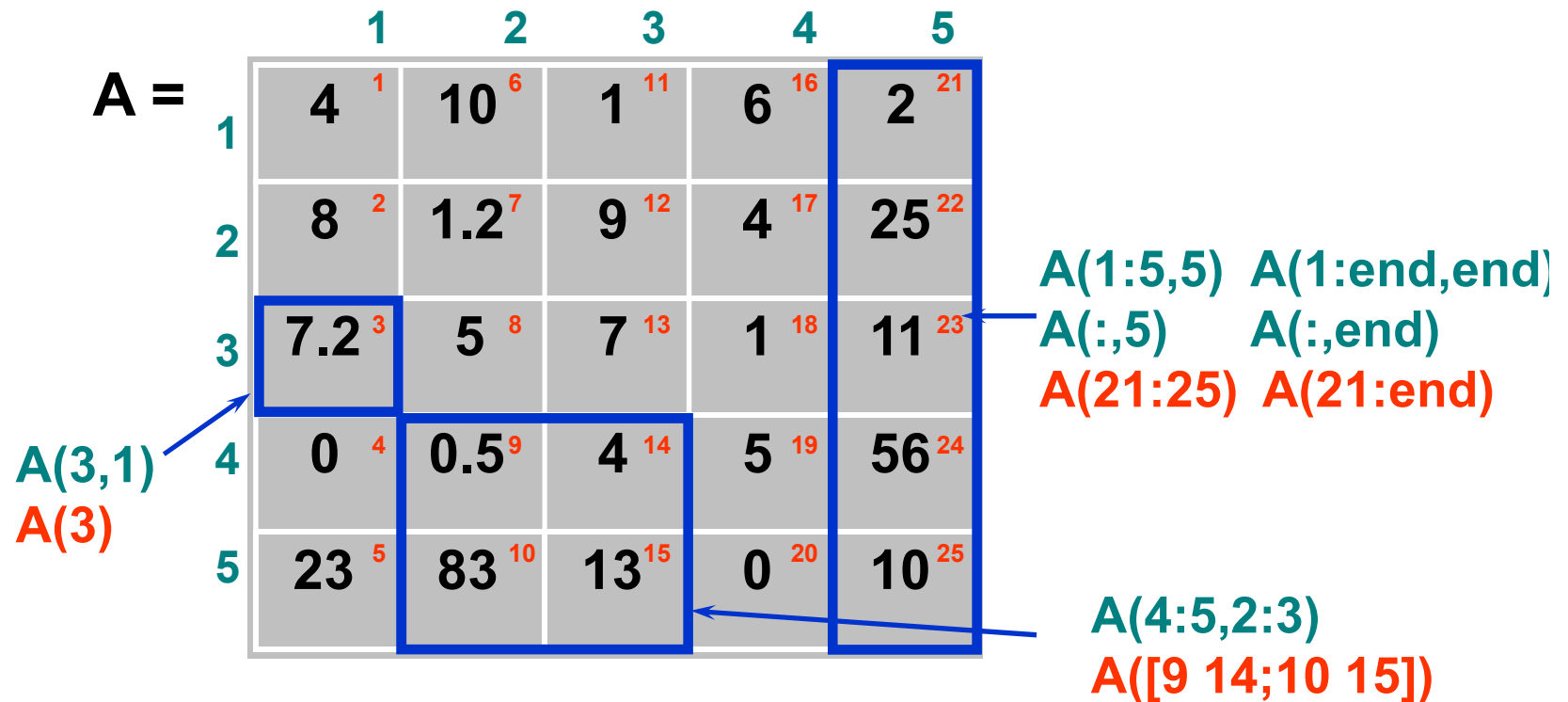
Error:

??? Error using ==> vertcat CAT arguments dimensions are not consistent.

Apagar linhas e colunas

```
» A=[1 5 9;4 3 2.5; 0.1 10 3i+1]
A =
    1.0000         5.0000         9.0000
    4.0000         3.0000         2.5000
    0.1000        10.0000    1.0000+3.0000i
» A(:,2)=[]
A =
    1.0000         9.0000
    4.0000         2.5000
    0.1000    1.0000 + 3.0000i
» A(2,2)=[]
??? Indexed empty matrix assignment is not allowed.
```


Array Subscripting / Indexing



Operadores (aritméticos)

- + adição
- subtracção
- * multiplicação
- / divisão
- ^ potência
- ' transposta ou complexa conjugado

Operações matriciais

Dados:
matrizes A e B

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

A =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

```
>> B = [3 5 2; 5 2 8; 3 6 9]
```

B =

| | | |
|---|---|---|
| 3 | 5 | 2 |
| 5 | 2 | 8 |
| 3 | 6 | 9 |

Adição

```
>> X = A + B
```

X =

| | | |
|----|----|----|
| 4 | 7 | 5 |
| 9 | 7 | 14 |
| 10 | 14 | 18 |

Subtracção

```
>> Y = A - B
```

Y =

| | | |
|----|----|----|
| -2 | -3 | 1 |
| -1 | 3 | -2 |
| 4 | 2 | 0 |

Produto

```
>> Z = A * B
```

Z =

| | | |
|----|-----|-----|
| 22 | 27 | 45 |
| 55 | 66 | 102 |
| 88 | 105 | 159 |

Transposta

```
>> T = A'
```

T =

| | | |
|---|---|---|
| 1 | 4 | 7 |
| 2 | 5 | 8 |
| 3 | 6 | 9 |

Operadores (Elemento por Elemento)

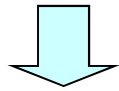
- .^{*} multiplicação elemento-por-elemento
- .[/] divisão elemento-por-elemento
- .[^] potênciação elemento-por-elemento

O uso de “.” – Operação “Elemento”

```
A = [1 2 3; 5 1 4; 3 2 1]
```

A =

```
1 2 3
5 1 4
3 2 -1
```



```
x = A(1,:)
```

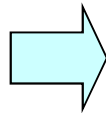
x =

```
1 2 3
```

```
y = A(3 ,:)
```

y =

```
3 4 -1
```



```
b = x .* y
```

b =

```
3 8 -3
```

```
c = x ./ y
```

c =

```
0.33 0.5 -3
```

```
d = x.^2
```

d =

```
1 4 9
```

```
K = x^2
```

Error:

??? Error using ==> mpower Matrix must be square.

```
B = x*y
```

Error:

??? Error using ==> mtimes Inner matrix dimensions must agree.

Multiplicação de Matrizes

```
» a = [1 2 3 4; 5 6 7 8];           [2x4]
» b = ones(4,3);                     [4x3]
» c = a*b                             [2x4]*[4x3] → [2x3]
c =
    10    10    10
    26    26    26
```

← a(2ª linha).b(3ª coluna)

Multiplicação de Matrizes Ponto-a-Ponto

```
» a = [1 2 3 4; 5 6 7 8];
» b = [1:4; 1:4];
» c = a.*b
c =
     1     4     9    16
     5    12    21    32
```

← c(2,4) = a(2,4)*b(2,4)

Funções para manipulação de Matrizes

- **zeros**: Cria uma matriz com todos os elementos zeros
- **ones**: Cria uma matriz com todos os elementos “1”
- **eye**: Matriz Identidade
- **rand**: Números aleatórios com distribuição uniforme em $[0,1]$
- **randn**: Números aleatórios com distribuição normal (mean=0 e std=1);
- **diag**: Matriz diagonal matrizes e diagonal de uma matriz
- **size**: Retorna a dimensão de uma matriz
- **fliplr**: “Flip” matrizes “left-right”
- **flipud**: “Flip” matrizes “up and down”
- **repmat**: Replica uma matriz

Funções para manipulação de Matrizes

- `transpose (')`: matriz transposta
- `rot90`: matriz rodada em 90°
- `tril`: parte triangular inferior de uma matriz
- `triu`: parte triangular superior de uma matriz
- `cross`: vector produto cruzado ou produto vectorial
- `dot`: producto interno
- `det`: determinante de uma matrix
- `inv`: Inversa de uma matriz
- `eig`: calcula os valores e os vectores próprios
- `rank`: dimensão (rank) de uma matriz

Matrizes de caracteres (Strings)

- Criar usando delimitador de aspas simples(')

```
» str = 'Hi there,'  
str =  
Hi there,  
» str2 = 'Isn't MATLAB great?'  
str2 =  
Isn't MATLAB great?
```

- Cada caracter é uma matriz elementar
(16 bits de memória por caracter)

str =

| | | | | | | | | |
|---|---|--|---|---|---|---|---|---|
| H | i | | t | h | e | r | e | , |
|---|---|--|---|---|---|---|---|---|

 ← 1x9 vector

- Os caracteres são indexados tal como nas matrizes numéricas

Tipo de Datos 'Char'

```
» c = ['hello'];  
» whos  
  Name      Size      Bytes Class  
  c         1x5        10 char array  
Grand total is 5 elements using 10 bytes  
  
» c(1)  
ans =  
h
```

```
» d = [c, ' again'];  
» d  
d =  
hello again  
»  
» b = ['hello'; 'again'];  
» size(b)  
ans =  
     2     5  
» b  
b =  
hello  
again  
»
```

Concatenação de Matrizes String

Utilizar [] como operador:

Cada linha dever ter o mesmo comprimento

Separador linha:
Ponto e virgula (;)

Separador coluna:
espaço / virgula (,)

Para strings de tamanho diferentes usar:

- `strvcat`

```
» str = 'Hi there, ' ;
» str1 = 'Everyone! ' ;
» new_str = [str, ' ', str1]
new_str =
Hi there, Everyone!
» str2 = 'Isn't MATLAB great? ' ;
» new_str2 = [new_str; str2]
new_str2 =
Hi there, Everyone!
Isn't MATLAB great?
```

Vectores 1x9 (pointing to `str` and `str1`)

vectores 1x19 (pointing to `new_str` and `str2`)

Matriz 2x19 (pointing to the resulting matrix in `new_str2`)

```
» new_str3 = strvcat(str, str2)
new_str3 =
Hi there,
Isn't MATLAB great?
```

Matriz 2x19 (zero padded) (pointing to the resulting matrix in `new_str3`)

Trabalhar com matrizes de “String”

- Comparação de “Strings”
 - ❑ `strcmp`: compara frases
 - ❑ `strncmp`: compare os primeiros ‘N’ caracteres
 - ❑ `findstr`: encontra uma substring (subfrases) dentro de uma frase.
- Converte matrizes de números & string :
 - ❑ `num2str`: converte de matrizes numéricas para string.
 - ❑ `str2num`: converte de matrizes de string para numérica
 - ❑ `char`: converte o código ASCII para um caracter.

Uint8 e Doubles

■ Double

□ Maiores funções MATLAB

- doubles como argumento de entrada

- return double

- e.g.,

- Necessidade de converter 'uint8' para 'double' antes de realizar operação matemática

```
» a = 1:10
a =
1  2  3  4  5  6  7  8  9 10
» b = uint8(a)
b =
1  2  3  4  5  6  7  8  9 10
» whos
Name      Size      Bytes Class
a         1x10      80 double array
b         1x10      10 uint8 array

» b*2
??? Error using ==> *
Function '*' not defined for variables of class 'uint8'.

» (double(b))*2

ans =

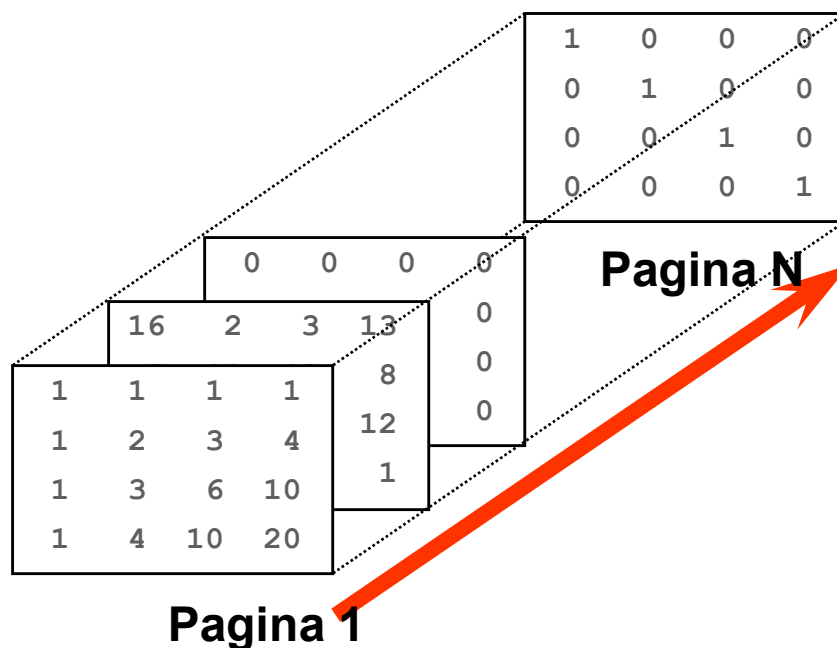
2  4  6  8 10 12 14 16 18 20
```

Matrizes Multidimensionais«

A 1ª dimensão da matriz é a linha - row.

A 2ª dimensão da matriz é a coluna - column.

A 3ª dimensão da matriz é a página- the *page*.



```
>> A = pascal(4);  
>> A(:, :, 2) = magic(4)  
A(:, :, 1) =  
    1    1    1    1  
    1    2    3    4  
    1    3    6   10  
    1    4   10   20  
  
A(:, :, 2) =  
   16    2    3   13  
    5   11   10    8  
    9    7    6   12  
    4   14   15    1  
  
>> A(:, :, 9) =  
    diag(ones(1, 4));
```

Estruturas / Structures

- Variável contendo campos (*fields*) de variáveis de dados com designações e tipos de dados diferenciados.

patient

.name

'John Doe'

.billing

127.00

.test

| | | |
|-----|-----|-------|
| 79 | 75 | 73 |
| 180 | 178 | 177.5 |
| 220 | 210 | 205 |

>> patient.name='John Doe';

>> patient.billing = 127.00;

>> patient.test= [79 75 73;
180 178 177.5;
220 210 205];

- A função *struct* cria uma matriz estrutura.
- Matriz de *structures*

```
>> patient(2).name='Katty Thomson';  
>> Patient(2).billing = 100.00;  
>> Patient(2).test= [69 25 33; 120 128 177.5; 220  
210 205];
```

Tipo de Dados 'Struct'

```
image.name = 'Tom';  
» image.height = 3;  
  
» image.width = 3;  
  
» image.data = [8 10 2; 22 7 22; 2 4 7];  
  
» whos  
Name      Size      Bytes Class  
  
image     1x1        590 struct array  
  
Grand total is 18 elements using 590 bytes
```


Tipo de dados Arrays de Estruturas

```
» image(1) = image;  
» image(2).name = 'Mary'  
» image(2).width = 4;  
» image(2).height = 4;  
» whos
```

| Name | Size | Bytes | Class |
|-------|------|-------|--------------|
| image | 1x2 | 894 | struct array |

Grand total is 28 elements using 894 bytes

```
» image
```

```
image =
```

```
1x2 struct array with fields:
```

```
name  
height  
width  
data
```

Arrays de Estruturas

```
» image(1) = image;  
» image(2).name = 'Mary';  
» image(2).width = 4;  
» image(2).height = 4;  
» whos
```

| Name | Size | Bytes | Class |
|-------|------|-------|--------------|
| image | 1x2 | 894 | struct array |

Grand total is 28 elements using 894 bytes

```
» image
```

image =

1x2 struct array with fields:

**name
height
width
data**

```
» image(2)
```

ans =

**name: 'Mary'
height: 4
width: 4
data: []**

```
» image(1)
```

ans =

**name: 'Tom'
height: 3
width: 3
data: [3x3 double]**

Cell Arrays

- Matriz cujos elementos são células (*cells*). Pode conter outras matrizes de diferentes tipos.

```
>> A(1,1) = {[1 4 3;  
0 5 8;  
7 2 9]};  
>> A(1,2) = {'Ana Simões'};  
>> A(2,1) = {3+7i};  
>> A(2,2) = {-pi:pi/10:pi};
```

| | |
|---|--------------------------------|
| cell 1,1 1 4 3 0 5 8 7 2 9 | cell 1,2 Ana Simões |
| cell 2,1 3+7i | cell 2,2 [-pi:pi/10:pi] |

- Utilizar chavetas `{}` para apontar aos elementos matriz célula
- Utilizar a função *celldisp* para mostrar a matriz célula

Tarefas Basicas: Desenhar a função $\sin(x)$ entre $0 \leq x \leq 4\pi$

- Criar uma matriz x com 100 amostras entre 0 e 4π .

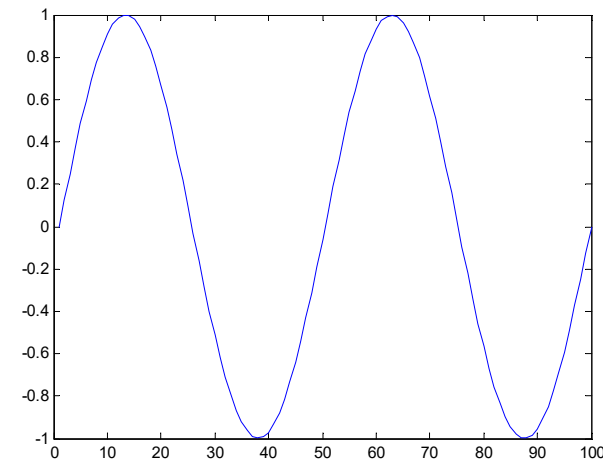
```
>>x=linspace(0,4*pi,100);
```

- Calcular $\sin(.)$ da array x

```
>>y=sin(x);
```

- Desenhar (Plot) a matriz y

```
>>plot(y)
```



Desenhar a função $e^{-x/3}\sin(x)$ entre $0 \leq x \leq 4\pi$

- Criar uma matriz x com 100 amostras entre 0 e 4π .

```
>>x=linspace(0,4*pi,100);
```

- Calcular $\sin(\cdot)$ da matriz x

```
>>y=sin(x);
```

- Calcular $e^{-x/3}$ da matriz x

```
>>y1=exp(-x/3);
```

- Multiplicar a matriz y com y1

```
>>y2=y*y1;
```

ERRADO !!!!!

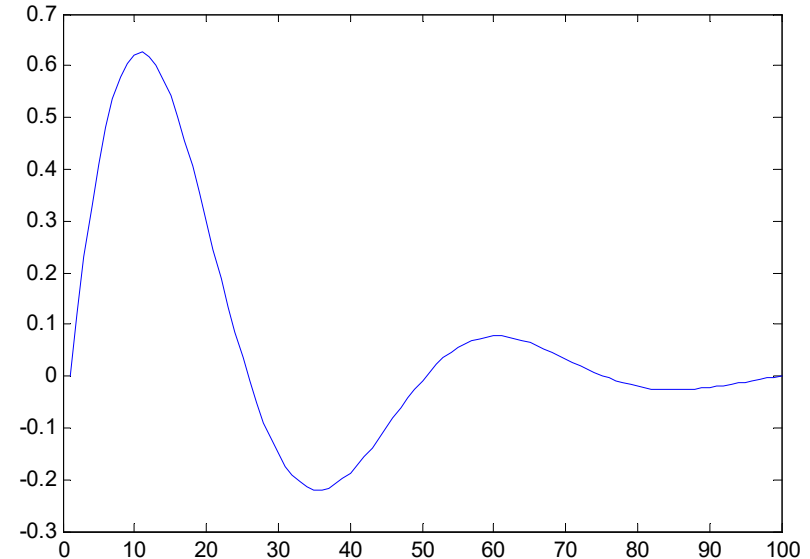
Desenhar a função $e^{-x/3}\sin(x)$ entre $0 \leq x \leq 4\pi$

- Multiplicar a matriz y e y1 **correctamente**

```
>>y2=y.*y1;
```

- Desenhar a matriz y2

```
>>plot(y2)
```

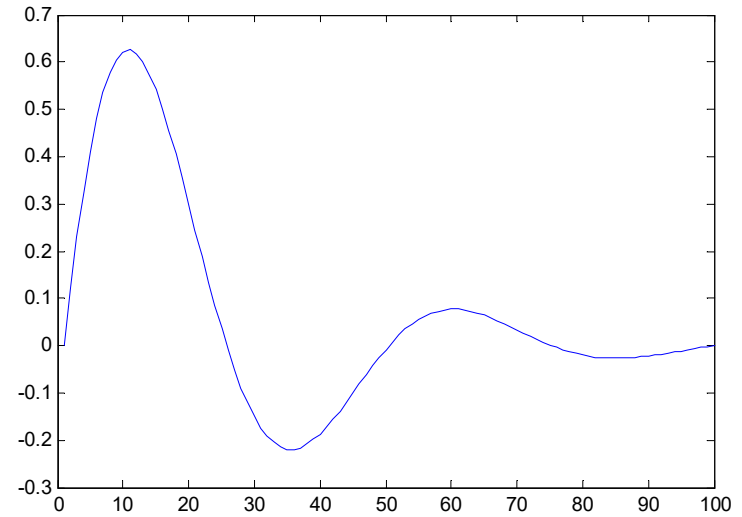


Opções de “Display”

■ plot(.)

Example:

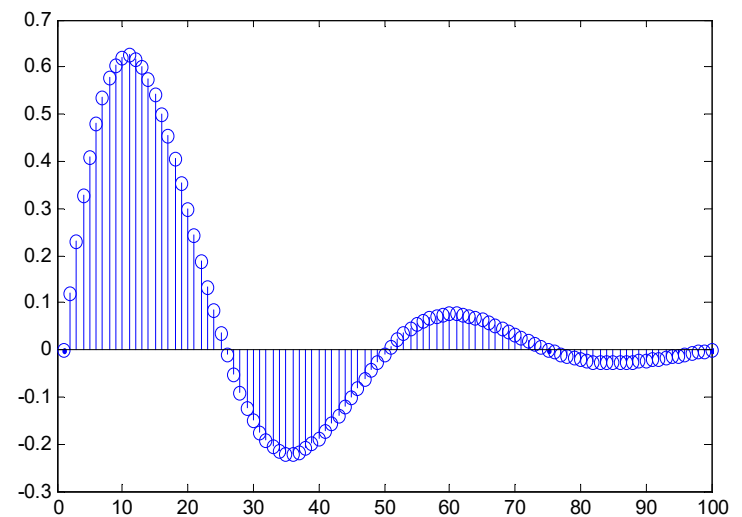
```
>>x=linspace(0,4*pi,100);  
>>y=sin(x);  
>>plot(y)  
>>plot(x,y)
```



■ stem(.)

Example:

```
>>stem(y)  
>>stem(x,y)
```



Opções de “Display”

■ title(.)

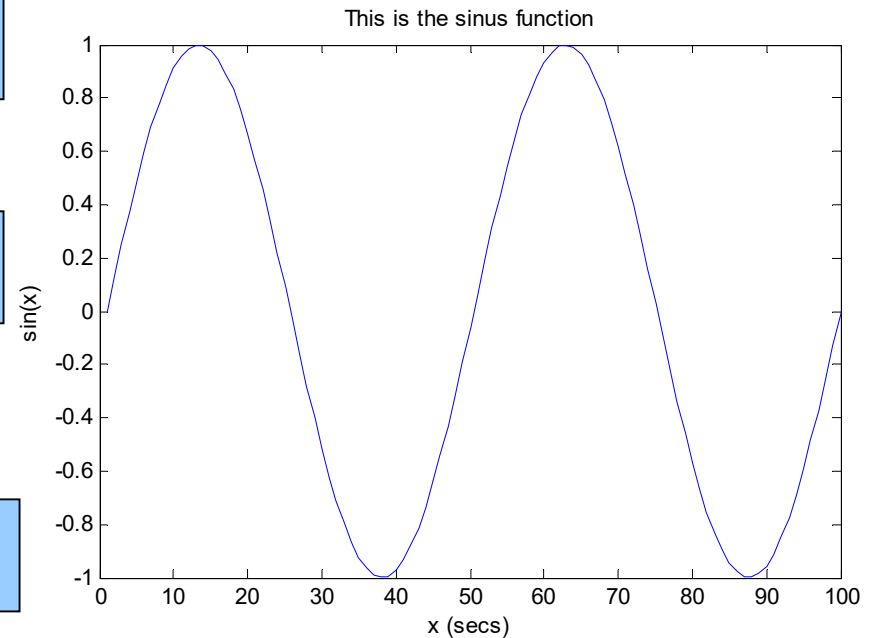
```
>>title('This is the sinus function')
```

■ xlabel(.)

```
>>xlabel('x (secs)')
```

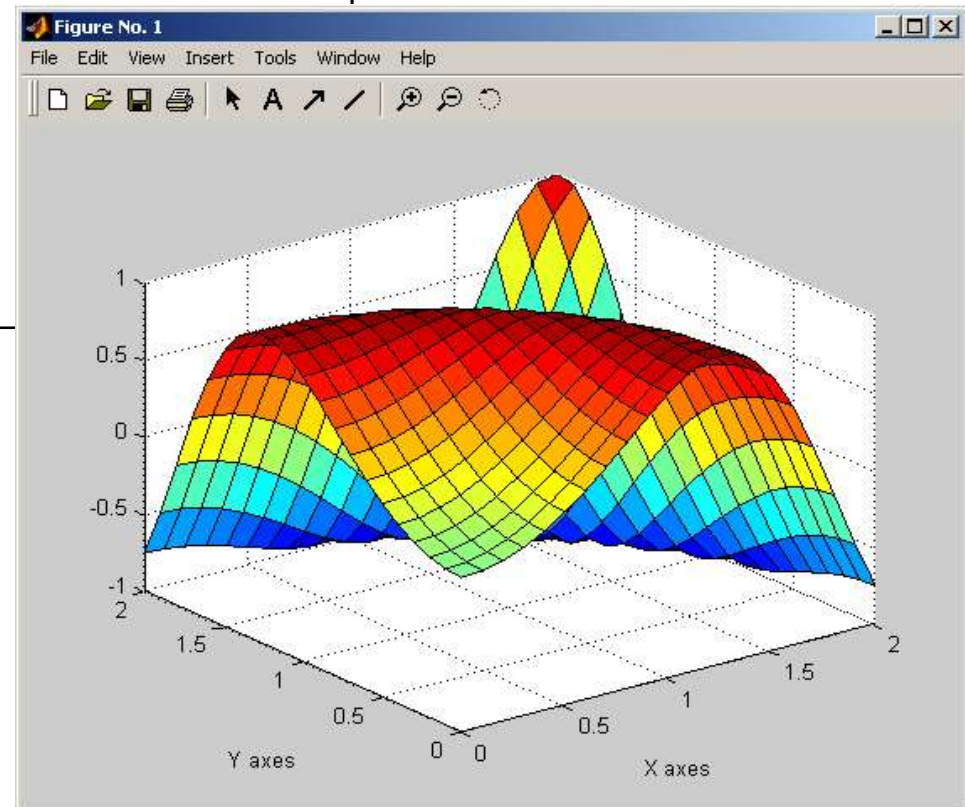
■ ylabel(.)

```
>>ylabel('sin(x)')
```



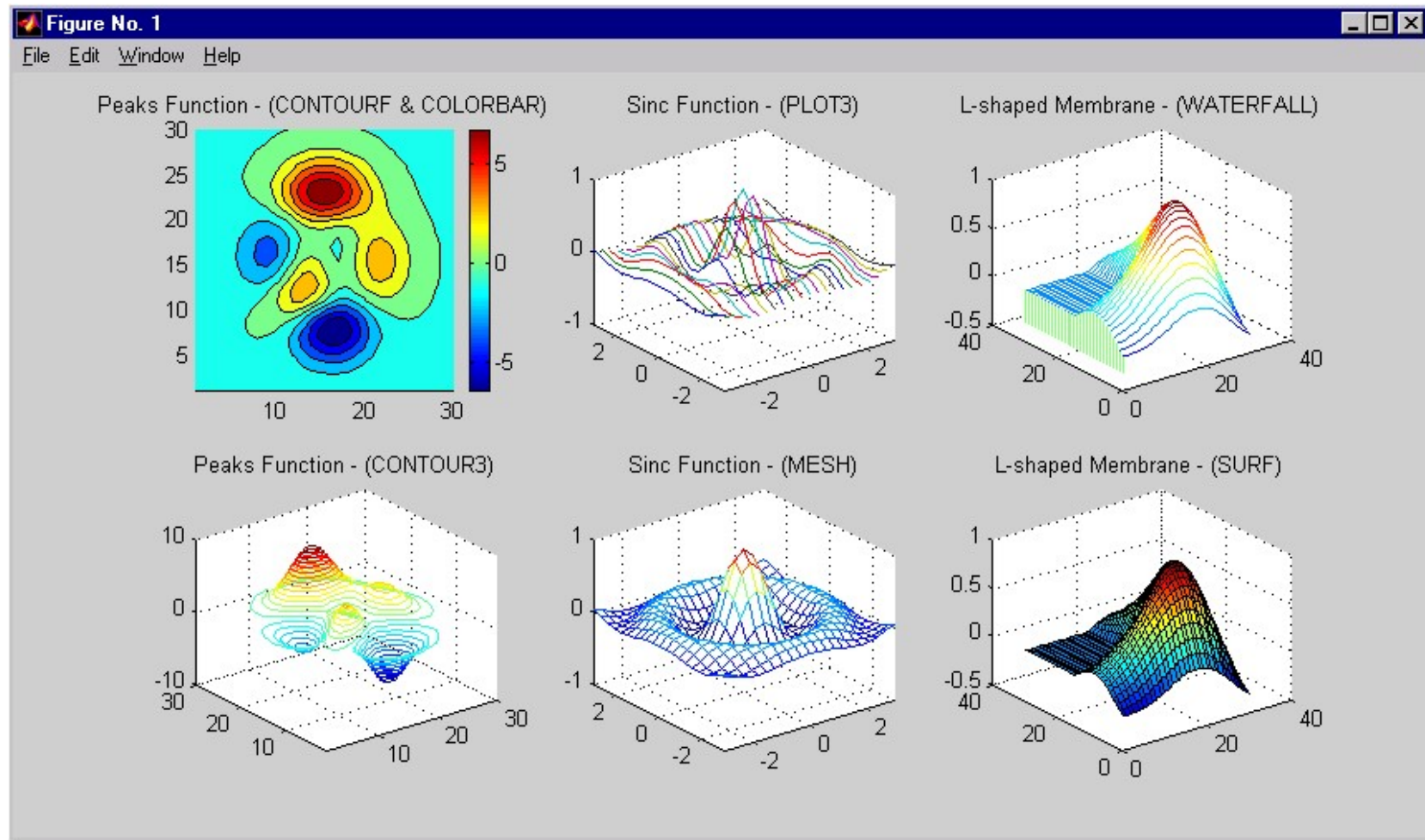
Surface Plot

```
x = 0:0.1:2;  
y = 0:0.1:2;  
[xx, yy] = meshgrid(x,y);  
zz=sin(xx.^2+yy.^2);  
surf(xx,yy,zz)  
xlabel('X axes')  
ylabel('Y axes')
```



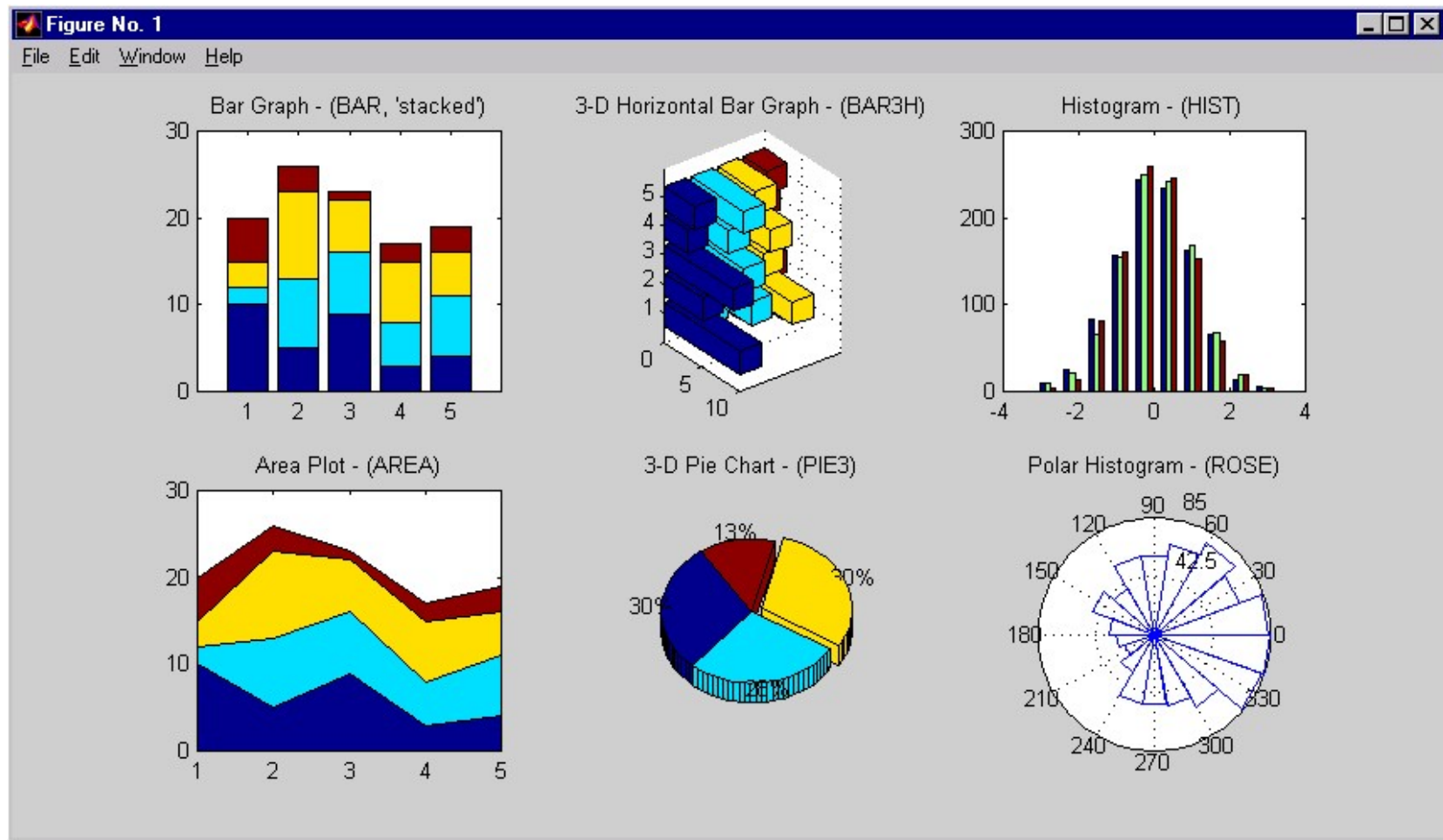
3 D Surface Plot

contourf-colorbar-plot3-waterfall-contour3-mesh-surf



Rotinas especializadas de “Plotting”

bar-bar3h-hist-area-pie3-rose



Operadores (relacionais, logicos)

- == Equal to (igual a)
- ~= Not equal to (diferente a)
- < Strictly smaller (menor que)
- > Strictly greater (maior que)
- <= Smaller than or equal to (menor ou igual que)
- >= Greater than equal to (maior ou igual que)
- & And operator (operador “e”)
- | Or operator (operador “ou”)

Controlo de Fluxo

- if
- for
- while
- break
-

Estruturas de Controlo

■ If Statement Syntax

```
if (Condition_1)
    Matlab Commands
elseif (Condition_2)
    Matlab Commands
elseif (Condition_3)
    Matlab Commands
else
    Matlab Commands
end
```

Alguns exemplos

```
if ((a>3) & (b==5))
    Comandos do Matlab;
end
```

```
if (a<3)
    Comandos do Matlab;
elseif (b~=5)
    Comandos do Matlab;
end
```

```
if (a<3)
    Comandos do Matlab;
else
    Comandos do Matlab;
end
```

Estruturas de Controlo

■ For loop syntax

```
for i=Index_Array  
    Matlab Commands  
end
```

Alguns exemplos

```
for i=1:100  
    Comandos do Matlab;  
end
```

```
for j=1:3:200  
    Comandos do Matlab;  
end
```

```
for m=13:-0.2:-21  
    Some Matlab Commands;  
end
```

```
for k=[0.1 0.3 -13 12 7 -9.3]  
    Comandos do Matlab;  
end
```

Estruturas de Controlo

■ While Loop Syntax

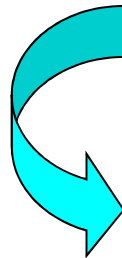
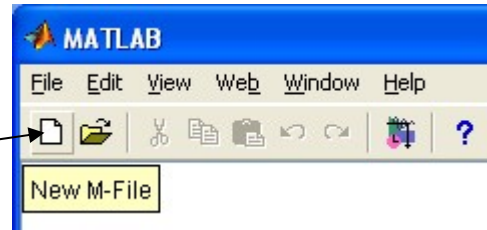
```
while (condition)
    Matlab Commands
end
```

Exemplo

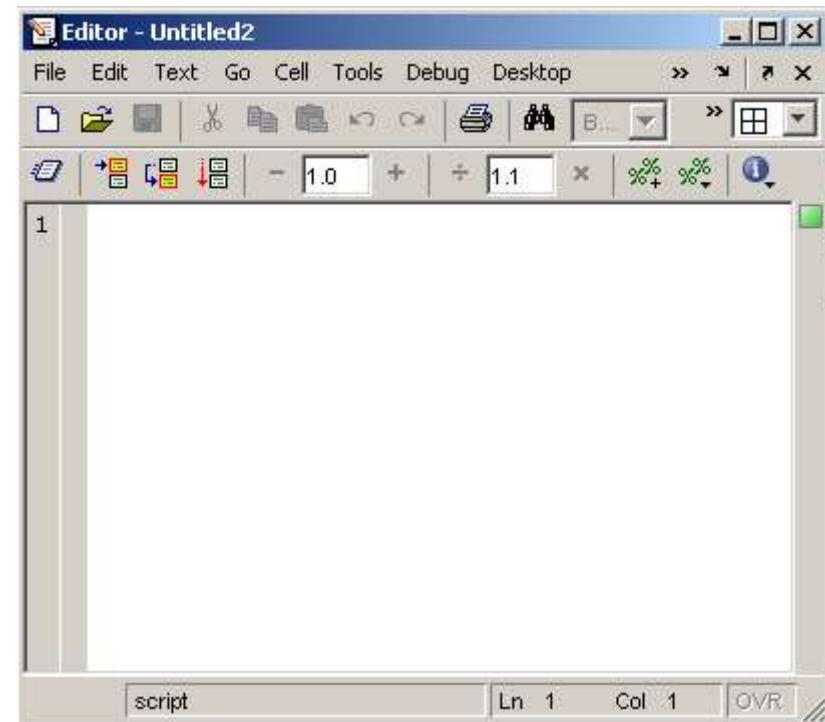
```
while ((a>3) & (b==5))
    Comandos do Matlab;
end
```


M-File

Click em “create a new M-File” (criar uma nova M-file)

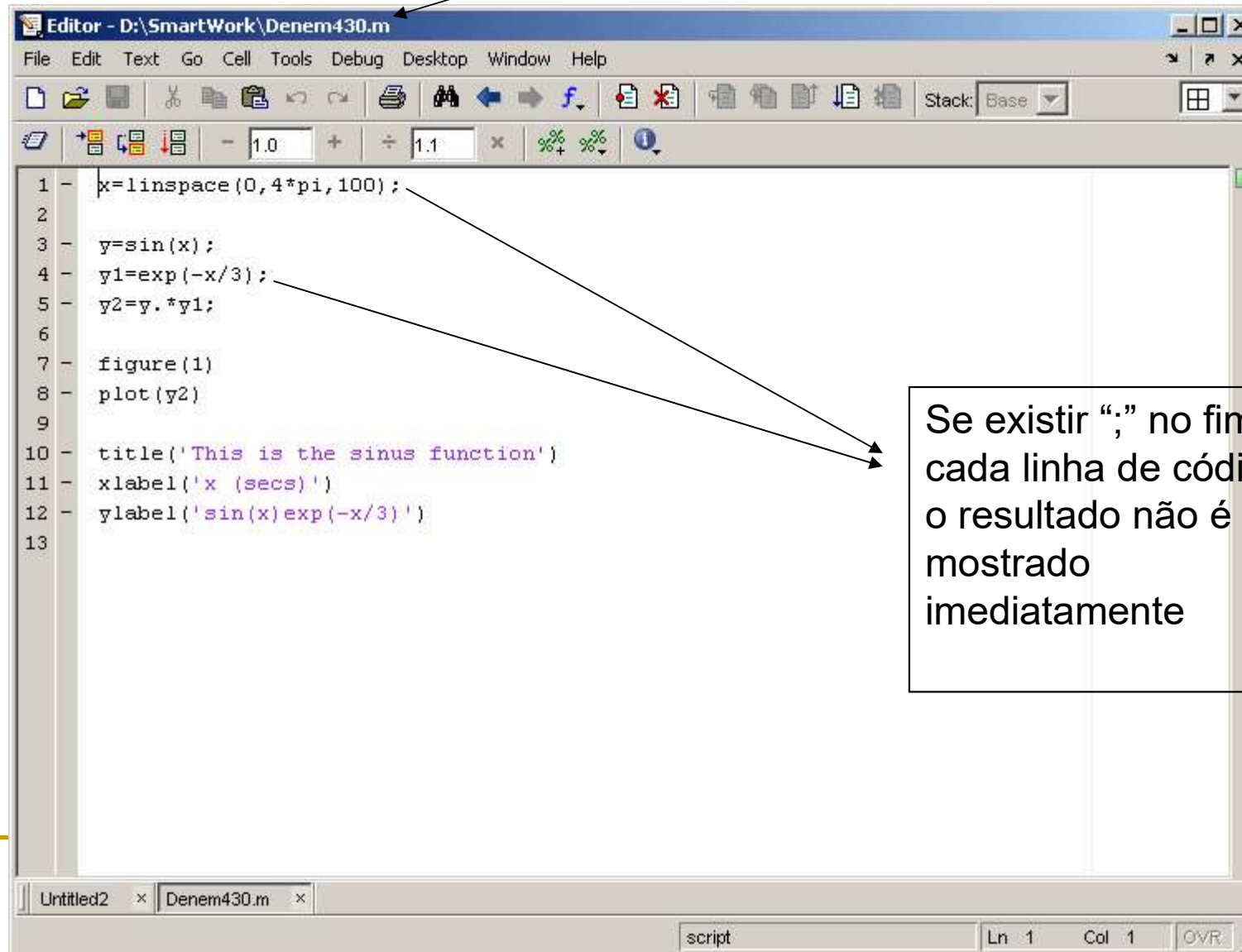


- Extensão “.m”
- Um ficheiro texto que contém “script”, funções ou programas para serem executados.



Uso da M-File

Save file as (guardar como) *Denem430.m*



Escrever uma função definida pelo utilizador

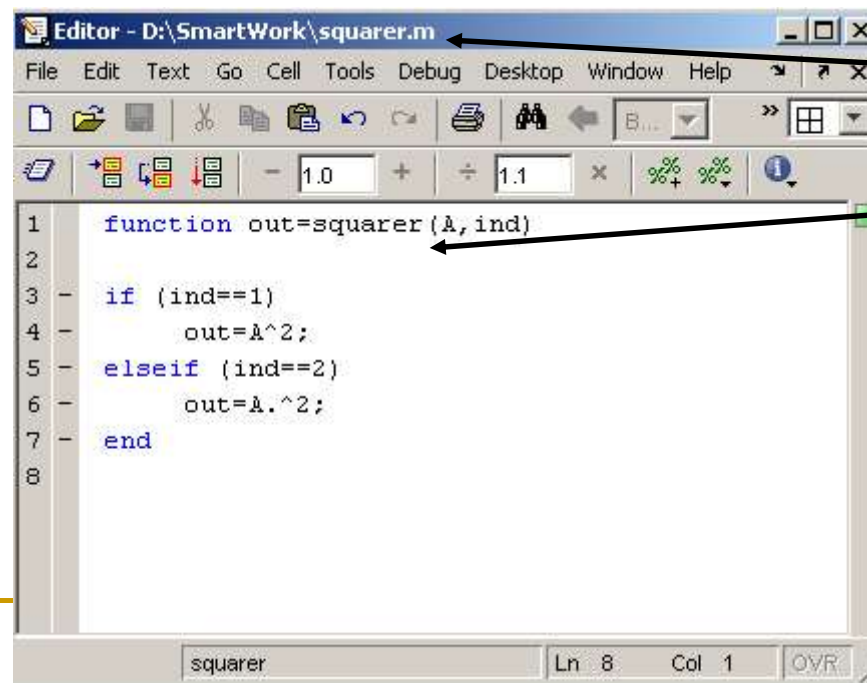
- As funções são m-files que podem ser executadas. Permitem a especificação de algumas entradas e devolução de saídas desejadas.
- A etiqueta de código “function” indica ao Matlab que se trata de uma função m-file:

```
function out1=functionname(in1)
function out1=functionname(in1,in2,in3)
function [out1,out2]=functionname(in1,in2)
```

- O comando “function” deve iniciar a m-file. Esta file deve ser gravada com o mesmo nome da função.

Escrever uma função definida pelo utilizador

- Exemplo
 - escrever a função : **out=squarer (A, ind)**
 - Deve devolver o quadrado da matriz de entrada se a entrada indicador (ind) for igual a 1
 - Deve devolver o quadrado elemento por elemento da matriz de entrada se o indicador (ind) for igual a 2



```
1 function out=squarer(A,ind)
2
3 if (ind==1)
4     out=A^2;
5 elseif (ind==2)
6     out=A.^2;
7 end
8
```

Com o mesmo nome

Notas:

- “%” As linhas de comentário são iniciadas pelo este caracter (equivalente ao “//” no C). Qualquer coisa escrita à sua direita nessa linha é negligenciada pelo compilador do Matlab.
- Por vezes é necessário (deliberadamente) abrandar a execução do código, frequentemente para permitir a observação de resultados intermédios. Para o efeito, utiliza-se o comando “pause”

```
pause %Espera até uma tecla ser pressionada  
pause(3) %espera 3 segundos
```



Exercícios

1 - Para as matrizes A e B definidas, execute as seguintes operações e interprete os resultados.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 3 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 1 \\ 3 & 7 & 0 \end{bmatrix}$$

- | | | | |
|--------------------------------------|----------|--------------------------|----------------------|
| a) $A+B$ | b) $A*B$ | c) $A.*B$ | d) $A.^2$ |
| e) A^2 | f) $A*A$ | g) $\text{inv}(A)$ | h) $\text{inv}(A)*A$ |
| i) $A \setminus A$ | j) A/A | k) $A/A - \text{eye}(3)$ | |
| l) $A*\text{inv}(A) - \text{eye}(3)$ | | | |

2 – Utilizando as funções *zeros*, *ones* e *eye* construa as seguintes matrizes:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

`zeros (4)`

`ones (3)`

`ones (2,1)`

`eye (2)`

3 – A partir das matrizes A e B do exercício 2 e $D = [A \ B]$ verifique quais das seguintes operações são possíveis.

a) $A * D$ b) $D * A$ c) $D' * A$ d) $A * D * B$

e) $A * B * D$ f) $A * D'$ g) $D(:, 1:3) * A$

h) $D(1:3, :) * A$

4 – A partir de um conjunto 1000 de valores aleatórios com distribuição normal (média 10 e desvio padrão 4) determine a percentagem de valores:

- a) superiores a 10
- b) entre 8 e 12;
- c) entre 6 e 14;
- d) entre 6 e 14;
- e) superiores a 15.

5- Represente graficamente a função de densidade de probabilidade da distribuição normal reduzida, no intervalo de -4 a 4.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

6 – A partir de um conjunto de 100 valores aleatórios com média 500 e desvio padrão 100, representando uma série de valores anuais de precipitação entre 1900 e 1999, elabore um programa que:

- a) Represente graficamente esta série temporal (Figura 1).
- b) Conte o número de ocorrências em que a precipitação excedeu o valor médio mais duas vezes o desvio padrão (valores considerados anómalos).
- c) Represente no gráfico, através de círculos pretos, os valores anteriores.
- d) Utilizando a função *hist*, construa um histograma, com 20 classes, que represente a distribuição da precipitação (Figura 2).

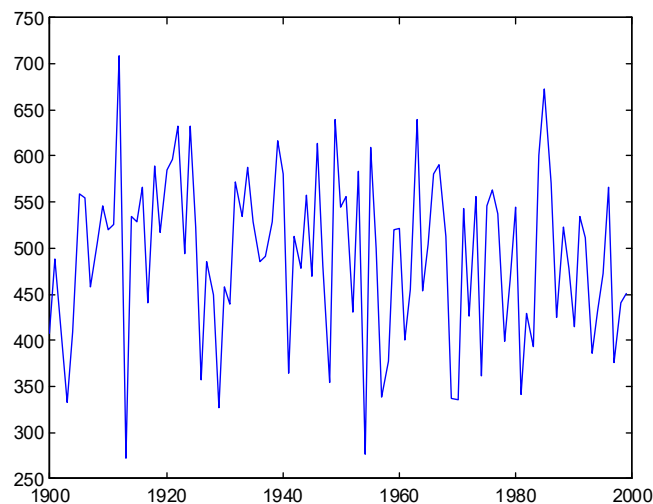


Figura 1

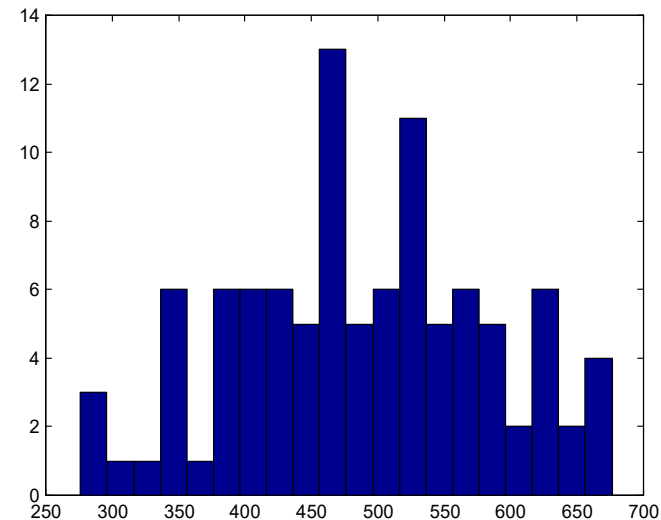


Figura 2

7 – Construa uma função que calcula o factorial de um número.

Questões

- ?
 - ?
 - ?
 - ?
 - ?
-



INTERVALO