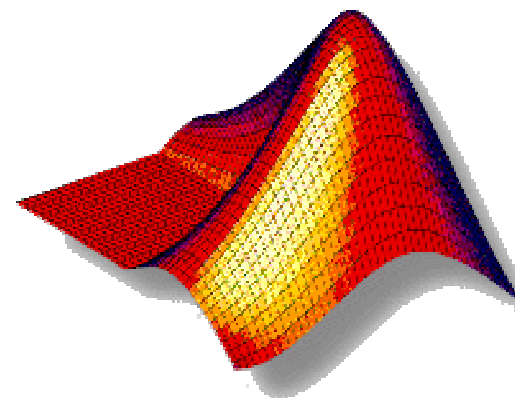


# MatLab

---

## Introdução



**Jorge Henriques,**  
**DEI, 2005**

---

# Índice

## 1. Em que consiste o MATLAB ?

## 2. Como usar o MATLAB?

- 2.1 Linha de comando
- 2.2 Uso de scripts

## 3. Tipos de Dados Fundamentais

- 3.1 Introdução/Definição de Matrizes
- 3.2 Referência a Elementos de Matrizes
- 3.3 Operações Elementares
- 3.4 Variáveis predefinidas
- 3.5 Escalares e Vectores

## 4. Graficação de Resultados

## 5. Controlo de Fluxo

- 5.1 Tipos de controlo

■ 5.2 Expressões Lógicas

**6. Ficheiros \*.m**

6.1 Modo Comando (on-line)

**7. Funções**

**8. Ficheiros Externos**

**9. Outros Comandos e Funções**

## 1. Em que consiste o MATLAB ?

O MATLAB consiste num pacote de funções orientadas fundamentalmente para o **cálculo científico**;

Apresenta-se sob a forma de um conjunto de bibliotecas (toolboxes), cada uma compreendendo funções específicas.

Exemplos:

- Neural Networks Toolbox
- Optimization Toolbox,
- Fuzzy toolbox
- ....

## 2. Como usar o MATLAB?

### 2.1 Linha de comando

A execução de comandos e funções pode ser efectuada directamente:

```
» plot(x,y)
```

### 2.2 Uso de scripts

- sucessão de funções e comandos reunidos num ficheiro de texto, obrigatoriamente de extensão \*.m

```
x= 1  
y= 2  
plot(x,y)  
sin(3*x)
```

## Comentários e final instrução

- % à direita da percentagem é entendido como um comentário
- ; no final de uma instrução impede a visualização do resultado

### 3. Tipos de Dados Fundamentais

**Em MatLab todos os dados são Matrizes**

Exemplo:

```
» x = 2+3
```

Os reais 2 e 3 são entendidos como duas matrizes de dimensão (1,1)

```
» A = [ 1 0 ; 0 1];
```

### 3.1 Introdução/Definição de Matrizes

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

De uma forma explícita

```
» A = [ 1 2 3 ; 4 5 6 ] ;
```

Resultado de alguma operação

```
» C = 3*A
```

```
» C = [ 3 6 9  
12 15 18 ]
```

LOAD de um ficheiro

```
» LOAD A
```



## 3.2 Referência a Elementos de Matrizes

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

### Um elemento

$A(i,j)$  - elemento linha  $i$ , coluna  $j$

```
» A(1,2)
```

```
» ans = 2
```

### Conjunto de elementos

$A(\text{imin:imax}, \text{jmin:jmax})$

```
» A(1:2, 2:3)
```

```
» ans =  $\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$ 
```

### 3.3 Operações Elementares

» $a^2$	- potenciação
» $A+B$	- adição
» $A*B$	- multiplicação
» $A/B$	- divisão
» $A'$	- transposta
» $\text{inv}(A)$	- inversa
» $\text{diag}(A)$	- diagonal
» $\text{ones}(m,n)$	- matriz de uns
» $\text{zeros}(m,n)$	- matriz de zeros
» $\text{eye}(m)$	- matriz identidade
» $\text{abs}()$	- valor absoluto
» $\text{min}()$	- mínimo,

Exemplos:

```
» ones(2,1)
```

```
» ans =  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 
```

```
» B= eye(3)
```

```
» B =  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 
```

### 3.4 Variáveis Pré-definidas

<code>ans</code>	Resultado (answer)
<code>pi</code>	Variável $\pi = 3,1416$
<code>inf</code>	Infinito
<code>nan</code>	not a number
<code>flops (obsoleta)</code>	Número de operações (em vírgula flutuante)
<code>eps</code>	Distância de um número ao real mais próximo
<code>realmin</code>	Menor número real
<code>realmax</code>	Maior número real
<code>i</code> ou <code>j</code>	Imaginário

## 3.5 Escalares e Vectores

### Escalares

» **a=2**

### Vectores

$$f(x) = x^3 + 2x^2 + 3x + 1$$

```
» %--- definicao do polinomio
» f =[ 1 2 3 1 ]

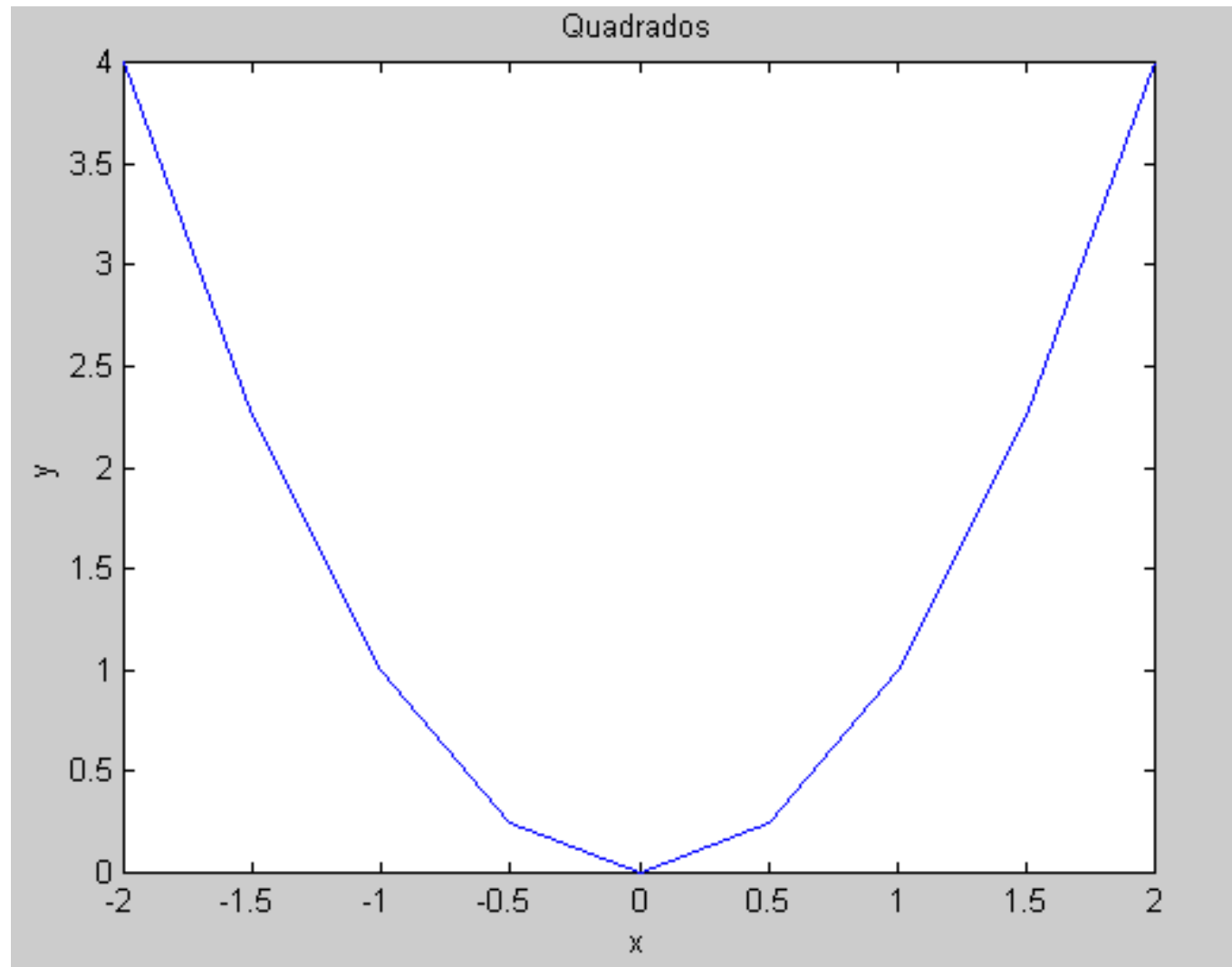
» %--- Raízes: solucao de f(x)=0
» roots(f)
» ans =  -0.7849 + 1.3071*j
          -0.7849 - 1.3071*j
          -0.4832
```

## 4. Graficação de Resultados

Funções mais utilizadas

- plot
- xlabel
- ylabel
- title

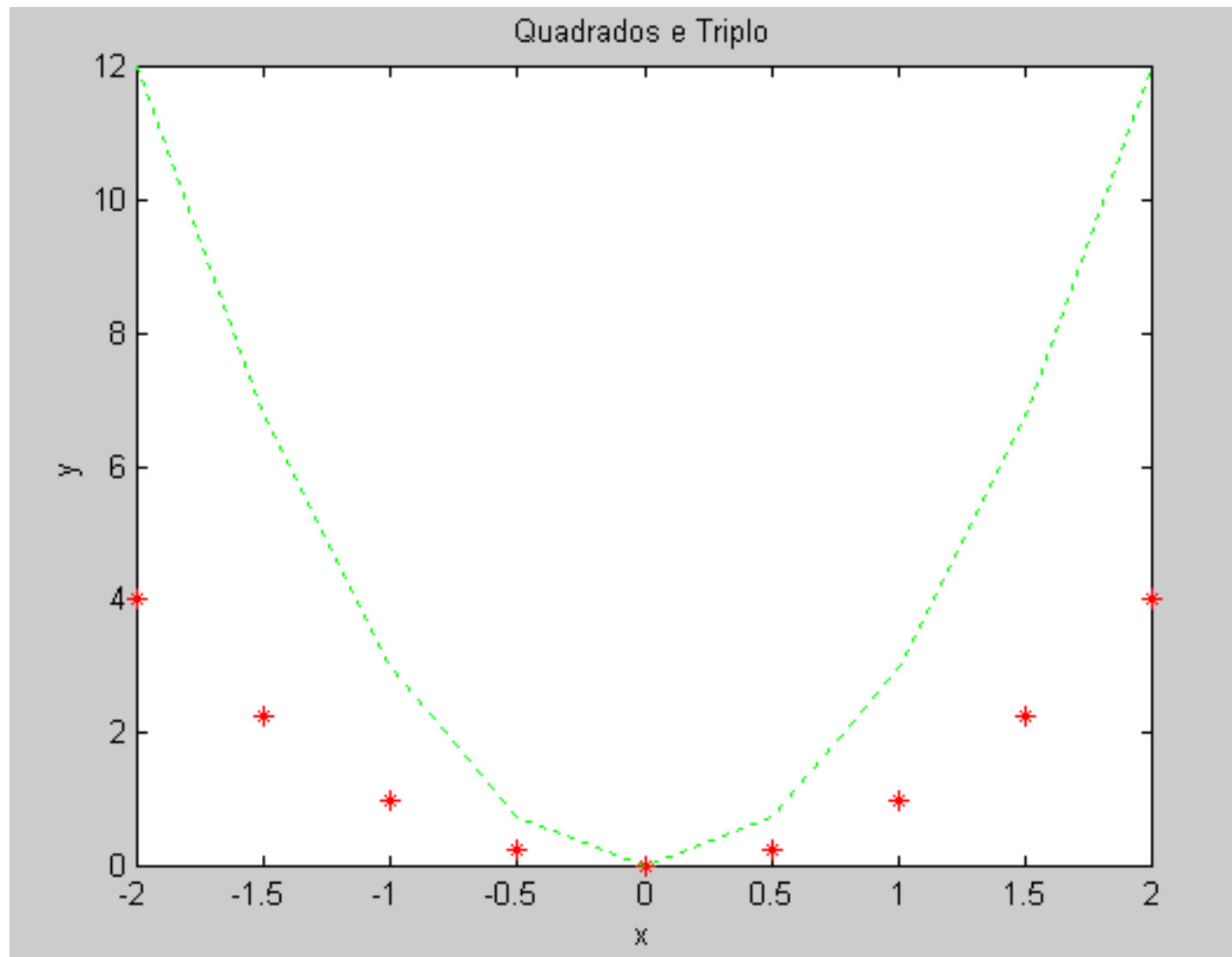
```
» %----- x= inicio:incremento:fim
» %----- y= x^2
» x = -2:0.5:2;
» y = x.^2;
» plot(x,y)
» title('Quadrados')
» xlabel('x')
» ylabel('y')
```



## Representação simultânea de vários gráficos

» `plot(t,y1, tipo linha, cor ...)`

```
» plot(x,y,'*r',x,3*y,':g')  
» title('Quadrados e Triplo')  
» xlabel('x')  
» ylabel('y')
```

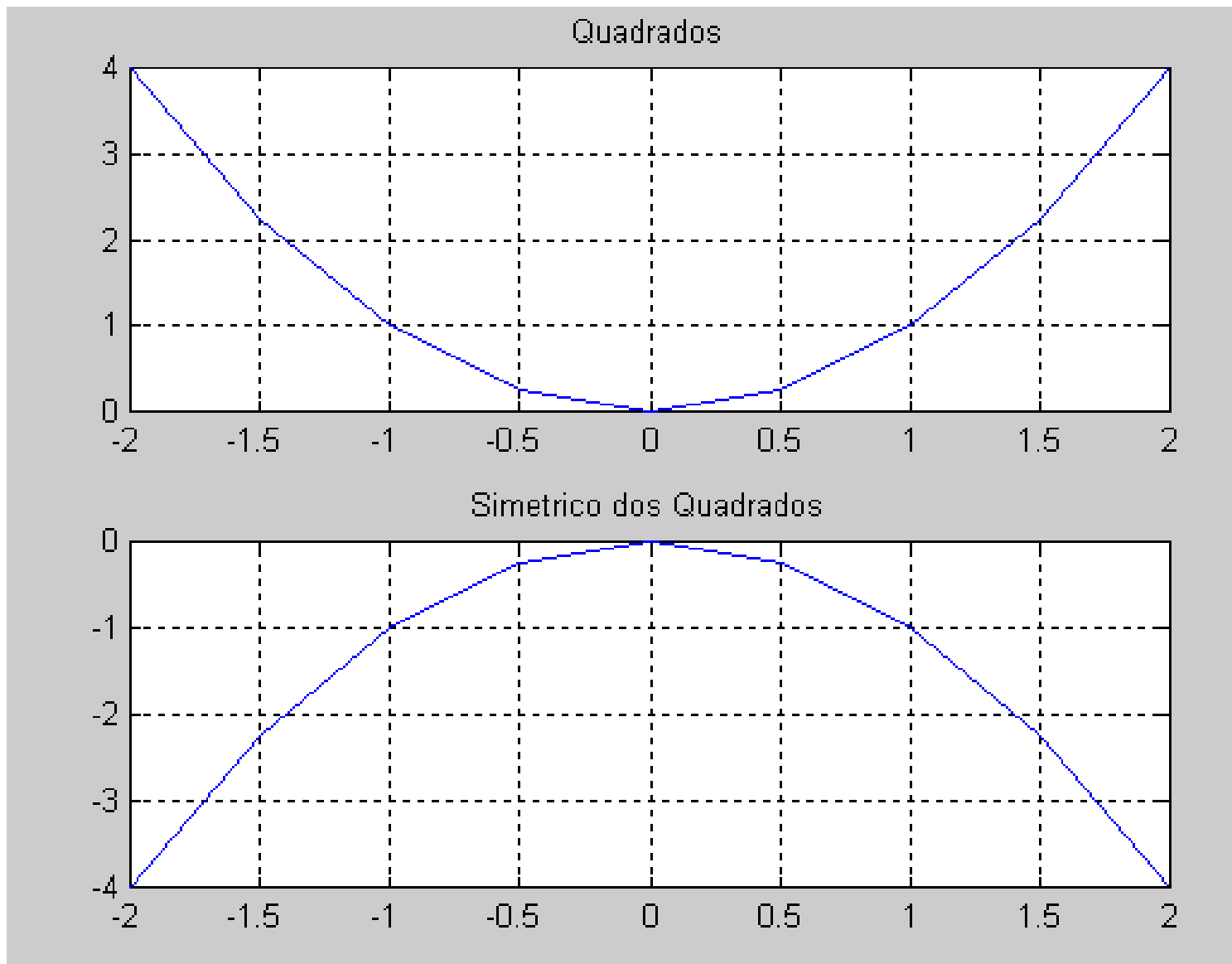




## Podem-se visualizar vários gráficos simultaneamente

» `subplot(colunas, linhas, grafico)`

```
» subplot(211)
» plot(x,y)
» grid
» title('Quadrados')
» subplot(212)
» plot(x,-y)
» grid
» title('Simetrico dos Quadrados')
```



## 5. Controlo de Fluxo

### 5.1 Tipos de controlo

- **for**
- while**
- if elseif**
- break**

**for**

```
>> for var=inicio:incremento:fim  
    .  
    .  
>> end
```

## while

```
» while expressao  
  .  
  .  
» end
```

## if

```
» if expressão  
  .  
  .  
» elseif expressão  
  .  
  .  
» end
```

## 5.2 Expressões Lógicas

■	<	menor	>	maior
	=	igual	~	diferente
	&	AND		OR
	~	NOT		

### Exemplo

```
»% Contagem de numeros negativos e positivos
```

```
» positivo = 0; negativo = 0;  
» for i=1:N  
    if x(i)>0  
        positivo=positivo+1;  
    end;  
    if x(i)<0  
        negativo=negativo+1;  
    end;
```

## 6. Ficheiros \*.m

### 6.1 Modo Comando (*on-line*)

```
» a=2  
» b=3  
» c= 2*a - 3*b
```

## 6.2 Scripts (*off-line*)

MATLAB permite “**executar**” ficheiros de texto com extensão \*.m

Por exemplo para inicializar algumas variáveis poder-se-ia criar um ficheiro ASCII

- **inicia.m**

```
a=2  
b=3  
c=4
```

o comando

```
» inicia
```

é equivalente a:

```
» a=2  
» b=3  
» c=4
```

**A maioria das funções disponíveis `abs`, `plot`, . . . são ficheiros `*.m`**

**Exemplo:**

Criar o ficheiro de nome `grau2` e extensão `m` (`grau2.m`).

```
a= input('parâmetro a: ? ');  
b= input('parâmetro b: ? ');  
c= input('parâmetro c: ? ');  
p= [a b c];  
r= roots(p);
```

Podem-se agora calcular as raízes de um polinómio de segunda ordem invocando

» **grau2**



## 7. Funções

A implementação de funções permite estender o conceito de scripts, residindo a principal diferença no cabeçalho do ficheiro (primeira linha):

```
function  
[argumentos_saída]=nome_função(argumentos_entrada)
```

- O ficheiro será identificado por `nome_função.m`

### Exemplo:

```
» function media= media (a, b)  
»   media=(a+b)/2;
```

Pode ser usado como

```
» c = media(12,23)
```

Podem existir vários parâmetros de entrada/saída

```
» function [so,di]= somadife(a,b)
»         so=a+b;
»         di=a-b;
» end;
```

Pode ser usado como

```
» [soma,diferenca] = somadife (12,23)
```

## 8. Ficheiros Externos

O MATLAB permite armazenar dados de uma sessão como um ficheiro (*com um formato especial*)

- **save** variáveis nome\_ficheiro
- load** variáveis nome\_ficheiro

```
%-- cria variaveis.mat que contem a e b  
» save a,b variaveis  
» load variaveis
```

**save** variáveis nome\_ficheiro / ascii

## 9. Outros Comandos e Funções

■ who	variáveis existentes
%	comentário
pause	pausa
exit, quit	sair
help	

Help: Como obter ajuda?

Uma função específica

```
» help poly
```

**lookfor** é outro comando de utilidade pois permite encontrar funções a partir de uma palavra chave específica:

```
» lookfor polynomial
```