Matlab num Instante

José Vieira

Dep. de Eletrónica, Telecomunicações e Informática da
Universidade de Aveiro

Vectores

Matrizes

Sinais

Sumário

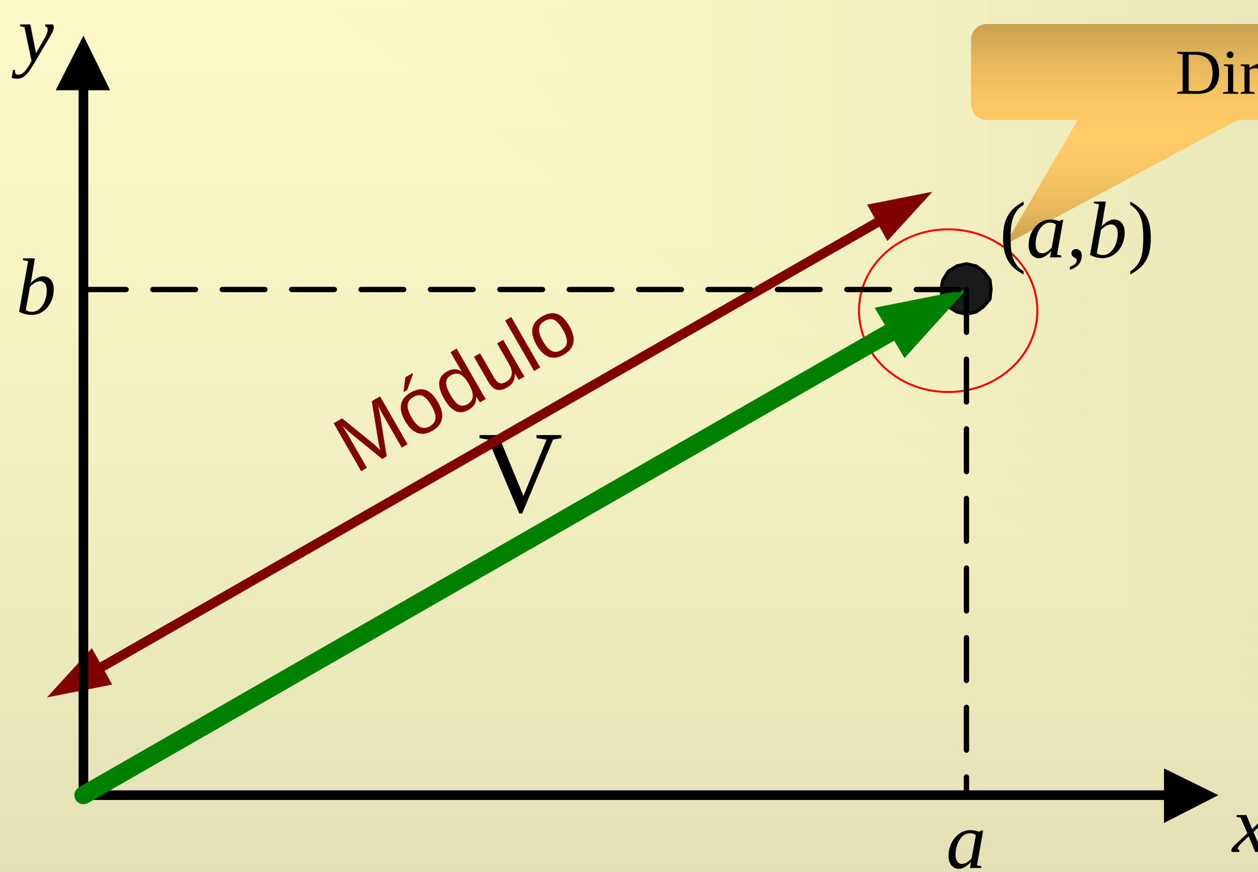
- Vectores
- Conceito geométrico de vector
- Indexação de matrizes.
- Transposta de uma matriz
- Sinais discretos como vectores de *N* dimensões
- Amostragem de sinais contínuos e a sua representação no Matlab como vectores.
- Matrizes como repositório de informação.

Sumário (cont.)

- Polinómios com o Matlab
- O operador “:”
- Manipulação de matrizes
- Sistemas de equações

Vectores

Conceito geométrico de vector (duas dimensões)



Vectores

- Da figura anterior pode-se concluir que bastam duas grandezas numéricas para representar um vector num espaço de duas dimensões.

$$(a,b)$$

Vectores

Num espaço com três dimensões são necessárias três grandezas:

$$(a,b,c)$$

Generalizando, um vector com N elementos pertence a um espaço com N dimensões.

Elementos de um espaço com mais de 3 dimensões são difíceis de representar graficamente.

Vectores

No Matlab para criar um vector "**v**" basta fazer por exemplo:

```
» v= [4, 5, 4, 2, 1, 7]
```

Os elementos são separados por espaços ou vírgulas

Matrizes

Uma matriz pode ser utilizada como um repositório de medidas

Exemplo:

Mede-se a temperatura do ar 4 vezes ao dia, durante uma semana, numa estação meteorológica.

Matrizes

Definição não rigorosa:

Uma matriz é uma tabela de números

Exemplos:

Vector linha

$$A \quad \begin{array}{cccc} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{array} \qquad v \quad \begin{array}{c} 1 \\ 3 \\ 1 \\ 7 \end{array} \qquad v \quad \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}$$

$$n \quad 7$$

Vector coluna

Matrizes - Índices

$$A \qquad \begin{array}{ccccc} 16 & 2 & 3 & 13 & 1 \\ 5 & 11 & 10 & 8 & 2 \\ 9 & 7 & 6 & 12 & 3 \\ 1 & 2 & 3 & 4 & \end{array}$$

$A_{2,3}$

Transposta de uma matriz

- A operação de transposição troca as linhas pelas colunas de uma matriz. Em notação matemática a transposta de uma matriz A representa-se por A^T. Em notação Matlab a transposta de uma matriz representa-se por **`A'`**
- Exemplo:

$$A \quad \begin{matrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{matrix} \qquad\qquad A^T \quad \begin{matrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{matrix}$$

Sinal

Os sinais podem ser classificados em

- Contínuos

  Um sinal diz-se contínuo se se puder medir o seu valor em qualquer instante de tempo

  Ex: a temperatura ambiente é um sinal contínuo

- Discretos

  Apenas se conhecem medidas do sinal tiradas em alguns instantes de tempo

  Ex: a temperatura ambiente medida todas as horas

Matrizes

Temperaturas registadas durante uma semana em 4 locais

$$T = \begin{matrix} 10 & 11 & 10 & 9 & 10 & 11 & 10 \\ 7 & 8 & 8 & 6 & 7 & 9 & 7 \\ 22 & 24 & 22 & 18 & 22 & 18 & 24 \\ 18 & 19 & 18 & 16 & 17 & 16 & 19 \end{matrix}$$

Matrizes

- A organização da informação na forma de uma matriz apresenta várias vantagens:
  - A informação fica organizada
  - Qualquer valor armazenado pode ser indexado de forma inequívoca
  - A representação abstracta de um conjunto de valores é compacta (no exemplo anterior basta apenas a letra *T* para representar 28 temperaturas)

Representação de polinómios

Um polinómio no Matlab pode ser representado no Matlab por um vector com os seus coeficientes. Vejamos um exemplo:

$$p(x) \quad 2x^3 \quad 3x \quad 9$$

Este polinómio representa-se no Matlab como:

```
p= [2, 0, -3, 9]
```

O termo nulo tem de ser representado de forma explícita

Operações com polinómios

Operação	Matlab
$p(x)+q(x)$	`p+q`
$p(x) \times q(x)$	`conv(p,q)`
raízes de $p(x)$	`roots(p)`
polinómio com as raízes r_1, r_2, ...	`poly(r)`
Valor do polinómio $p(x)$ para vários valores de x.	`polyval(p,x)`

O operador “:”

- O mais versátil operador do MATLAB
- Permite definir de forma compacta um conjunto de valores (vector) em progressão aritmética.

```
>> x = início: passo : fim

exemplo
>> x= 2:2:20
>> 2, 4, 6, 8, 10
```

Tipos de dados elementares

- Vectores numéricos

```
» x = 1:10;
» x = 1 + linspace(1,20,10)*j; % vector complexo
» y = (x >= 5)  % vector booleano
y =
     0   0   0   0   1   1   1   1   1   1
```

- Vectores de caracteres

```
»  x = ['c','h','a','r']
x = char
» x = ['char']
x = char
```

Definição funcional de matrizes

Quando se pretende criar uma matriz cujos elementos se podem relacionar facilmente, o Matlab possui as seguintes funções:

- zeros(N,M) gera uma matriz de zeros com *N* linha e *M* colunas
- ones(N,M) gera uma matriz de uns com *N* linha e *M* colunas
- rand(N,M) gera uma matriz de elementos aleatórios com *N* linha e *M* colunas
- magic(N) gera um quadrado mágico de dimensão *N*
- eye(N) gera uma matriz identidade de dimensão *N*

Exemplos

```
» A = eye(3)
A =
     1     0     0
     0     1     0
     0     0     1
```

```
» B = zeros(2,3)
B =
     0     0     0
     0     0     0
```

Concatenação

Com o Matlab é possível construir matrizes a partir de outras de menor dimensão. Eis alguns exemplos:

```
» x = [1 2; 3 4];
» A = [x x;x x]
A =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4

» % Problema de consistência
» x = [1 2 3 4; 4 5 6]
???  = [1 2 3 4; 4 5 6]
                       |
All rows in the bracketed expression must have the same
number of columns.
```

Dimensões

- Número de elementos dum vector ou matriz

```
» x = 1:10;
» y = 3 + j*linspace(1,10,20);
» dim_x = size(x), dim_y = size(y)
dim_x =
       1    10
dim_y =
       1    20
» A = rand(3,2);
» nelementos = prod(size(A));
» maior_dim = length(A); % maior dimensão matriz
» last = A(end,end); % Última linha, última coluna
```

Indexação

- Referência ao elemento *i,j* duma matriz

```
» A(3,2)
ans =
    0.7621
```

- O operador “:” revela-se um poderoso meio de indexação.

```
» x = 1:2:50;
» x(10:15)
ans =
    19    21    23    25    27    29
```

- Vectores de índices

```
» v1 = 10:15;
» x(v1)
ans =
    19    21    23    25    27    29
```

Índices lógicos

Em muitas situações pretende-se referenciar os elementos de uma matriz que satisfazem uma dada condição. Por exemplo, dado o vector

x= [1 2 –1 3 –3]

como se pode gerar um outro que apenas contenha os elementos menores que zero?

Se fizer **x<0** obtêm-se o seguinte vector lógico

0 0 1 0 1

Este vector pode ser utilizado para indexar os elementos de x

x(x<0)

-1 -3

Aritmética

- Soma algébrica com entidades escalares é extensível a vectores e matrizes desde que as dimensões sejam idênticas.

```
» A = rand(3);
» B = magic(3);
» C = A + B;
» C = A - B;
```

- Soma e multiplicação com valor escalar

```
» D = 5 + B; E = 5*B;
» F = 7 + 3*B – 12*A;
» F = (2 + 2j)*ones(3);
```

Multiplicação Aritmética

- Multiplicação aritmética ".*" ("elemento a elemento)

```
>> x = [1 2 3 4]; y = [2 2 10 10];

>> p = x .* y % Pointwise multiplication

p =
     2     4    30    40
```

Divisão e Sistemas de Equações

- Considere a seguinte equação com uma incógnita

$$ax = b$$

- Resolve-se fazendo

$$a^{-1}ax = a^{-1}b$$
$$x = a^{-1}b$$

Divisão e Sistemas de Equações

- Considere-se agora o sistema de equações

$$\begin{array}{lllll} 2u & v & w & & 5 \\ 4u & 6v & & & 2 \\ 2u & 7v & 2w & & 9 \end{array}$$

que se pode escrever na forma algébrica e resolver da mesma forma

$$\underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 4 & 6 & 0 \\ 2 & 7 & 2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}_{x} \quad \underbrace{\begin{bmatrix} 5 \\ 2 \\ 9 \end{bmatrix}}_{b} \qquad \mathbf{A}x \quad b \qquad x \quad \mathbf{A}^{1}b$$

Exemplos

- Resolução de um sistema de equações pelo método da eliminação Gaussiana utilizando divisão de matrizes

```
>> A = [2 1 1;4 -6 0;-2 7 2];
>> b = [5 -2 9]';
>> x = A\b  %Left Division
```

- Resolução de um sistema de equações pelo cálculo directo da inversa de uma matriz

```
>> X = inv(A)*b  %Inverse of A
```

Gráficos com o Matlab I

Sumário

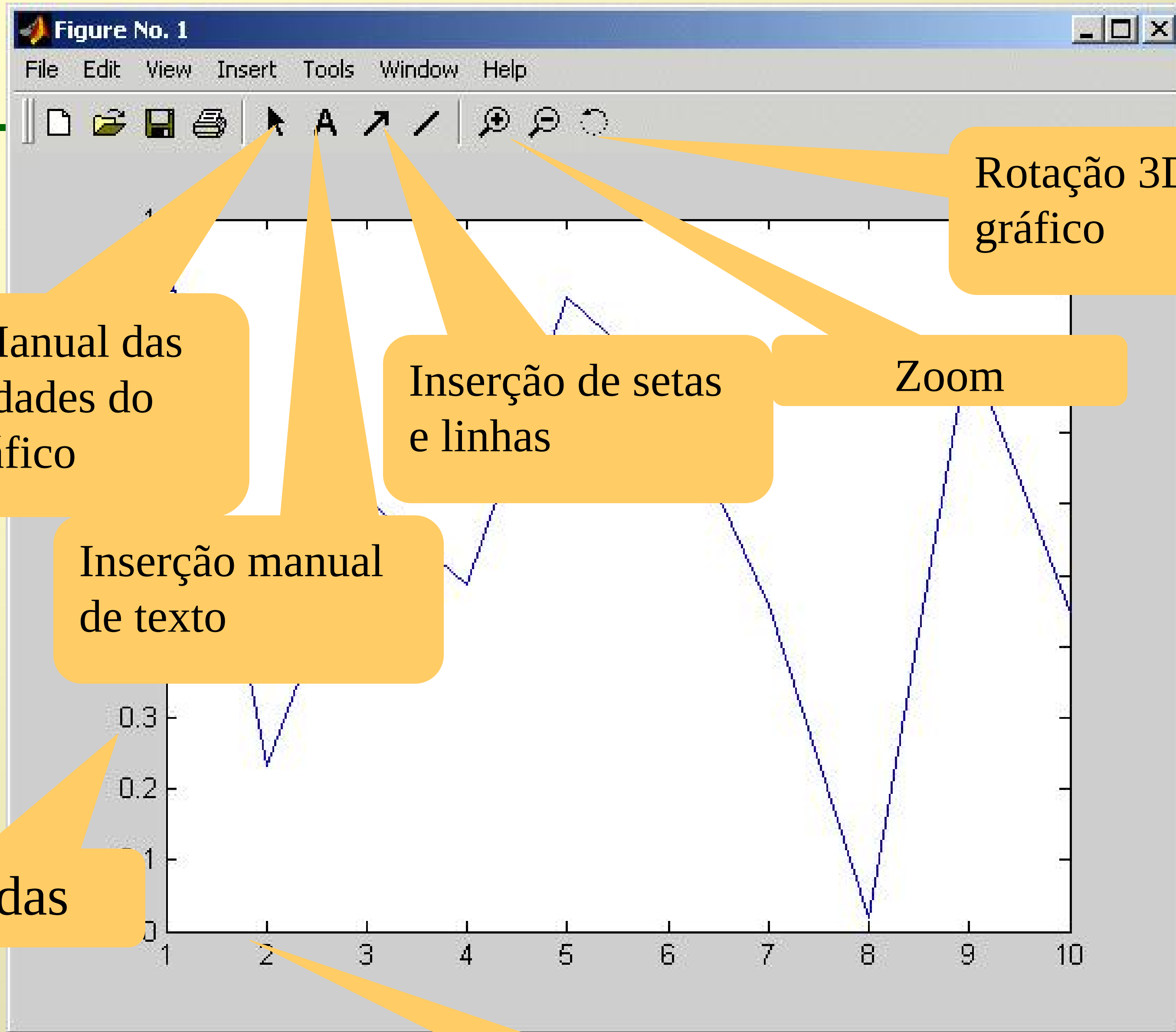
- Demonstração das potencialidades gráficas do Matlab
- Gráficos simples com o Matlab (plot)
  - Gráficos de uma variável
  - Alteração do aspecto de um gráfico
  - Inserção de texto num gráfico
  - Leitura de pontos de um gráfico
  - Eixos

Gráficos de uma Variável

- Sintaxe do comando **`plot`**

```
v= rand(1,10);
plot(v)
```

Nesta versão mais simples é desenhado um gráfico de linha contínua com a amplitude dos elementos do vector **`v`**. Nas abcissas aparecem os índices dos elementos de **`v`**.



Rotação 3D do gráfico

Edição Manual das propriedades do Gráfico

Inserção de setas e linhas

Zoom

Inserção manual de texto

Ordenadas

Abcissas

Sintaxe do comando **plot**

plot(x1,y1,x2,y2,....)

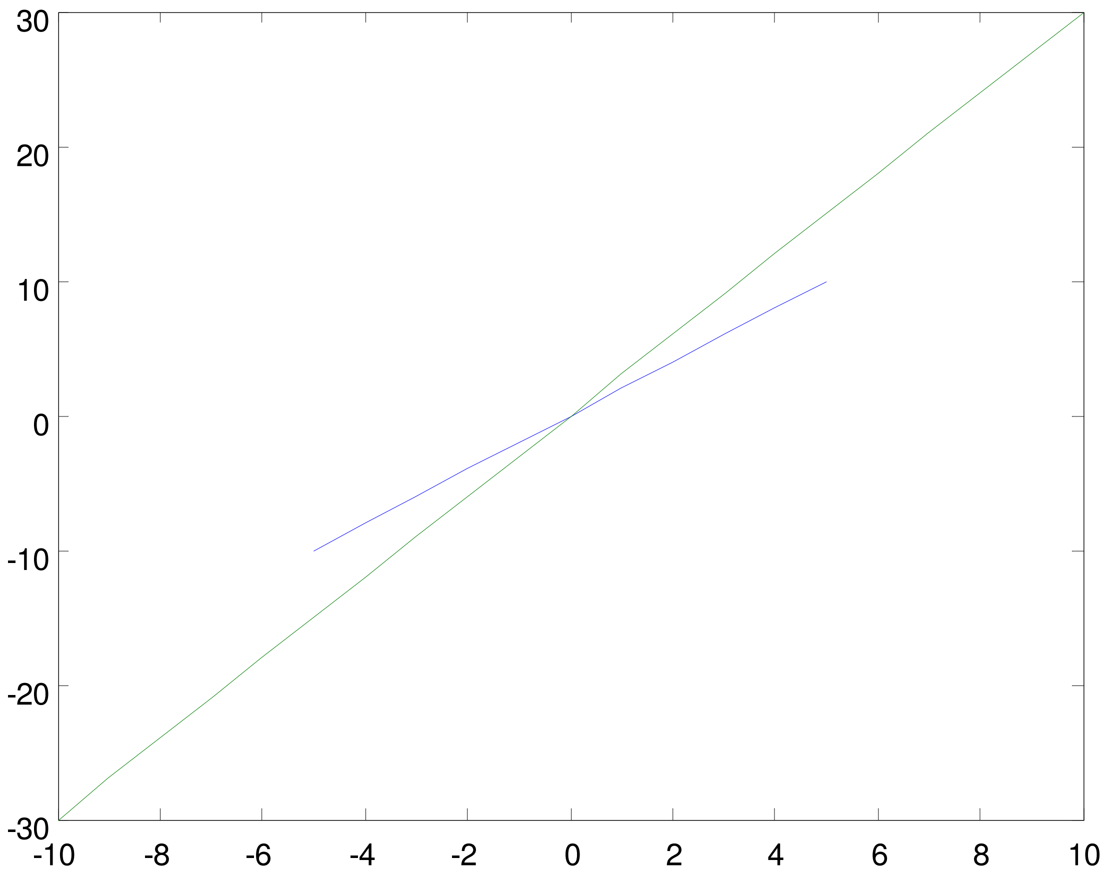
Os vectores das ordenadas **x1**, **x2**, ... podem ter um número diferente de elementos.

O número de elementos dos pares **(x1,y1)** e **(x2,y2)** deve ser o mesmo.

Exemplo:

```
x1= -5:5; x2= -10:10
y1= 2*x1; y2=3*x2;
plot(x1,y1,x2,y2)
```

Exemplo



Alteração do aspecto gráfico

Para além dos argumentos vectoriais a função plot permite ainda alterar o modo como as linhas são desenhadas. Essas indicações são codificadas na forma de uma "string" de texto colocada a seguir aos vectores dos pontos.

```
plot(x1,y1,'string1',x2,y2,'string2',...)
```

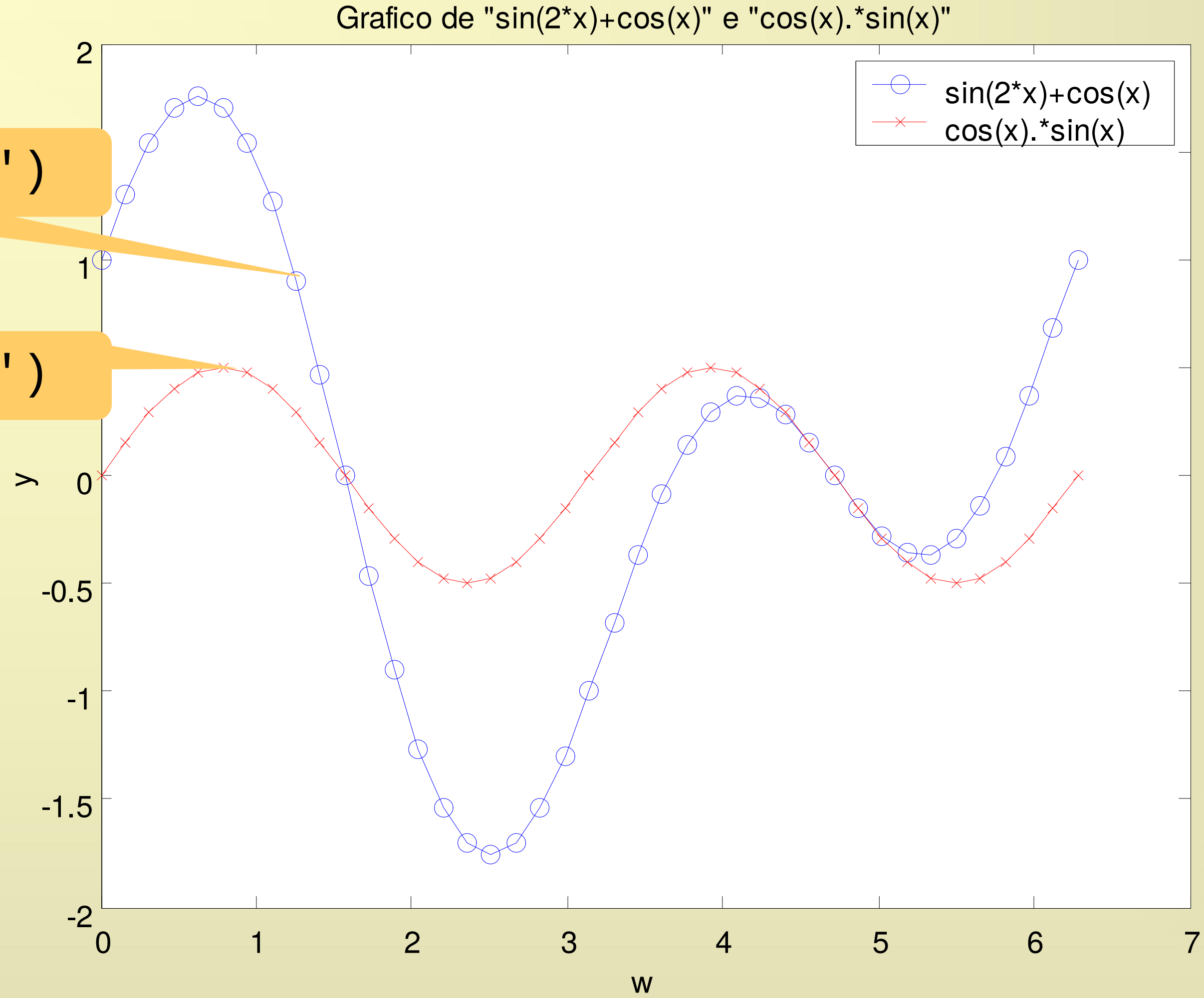
A "string" pode definir os seguintes atributos das linhas desenhadas

- Marcadores dos pontos do gráfico
- Cor das linhas e marcadores
- Tipo de linha a desenhar

Caractéres definidores de atributos

	Cor		**Marcadores**		**Linhas**
y	amarelo	.	ponto	-	linha a cheio
m	rosa	o	círculo	:	ponteada
c	azul claro	x	marca x	-.	traço ponto
r	encarnado	+	marca mais	--	tracejada
g	verde	*	estrela		
b	azul	s	quadrado		
w	branco	d	diamante		
k	preto	v	triângulo (cima)		
		^	triângulo (baixo)		
		<	triângulo (esquerda)		
		>	triângulo (direita)		
		p	pentagrama		
		h	“hexagram”		

Alteração do aspecto gráfico

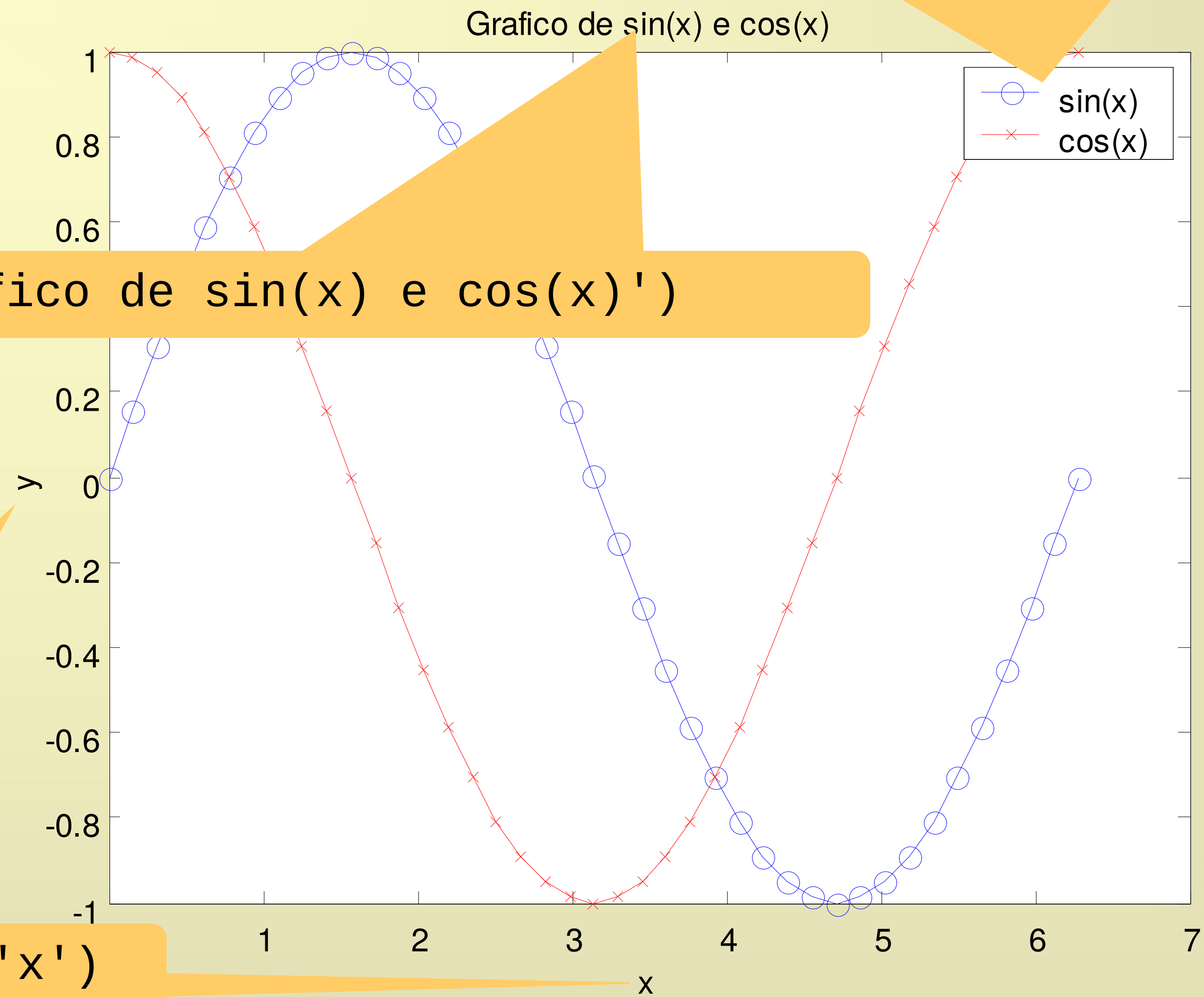



```
plot(x1,y1, '-ob')
```

```
plot(x2,y2, '-+r')
```

Legendas

```
legend('sin(x)','cos(x)')
```




```
title('Grafico de sin(x) e cos(x)')
```

```
ylabel('y')
```

```
xlabel('x')
```

Mínimos quadráticos

- Dados num ficheiro
- Carregar para Matlab
- Obter a reta que aproxima os dados

Animações

```
% Animação de uma sinusoide
clear all
t= 0:pi/50:4*pi;
figure(gcf)
y= sin(t);
plot(t,y)
axis off
a= get(gca,'children');
for n= 1:4
    for f= [1:0.1:4, 4:-0.1:1],
        y= sin(f*t);
        set(a,'XData',t,'YData',y,'Linewidth',2);
        drawnow;
        pause(0.01)
    end
end
```

Relatórios com o Matlab

```matlab
% Relatorios no Matlab

%% Parte 1 - Declaracao de variaveis

a= 1;
b= 2;

%% Parte 2 - Processamento das variaveis
% $c= a+b$

c= a+b;

```