

# Análise matemática C - PI

Rui Pereira

Departamento de Matemática para a Ciência e Tecnologia (DMCT) - Universidade do Minho

18 Abril de 2008

# MATLAB - resumo de algumas aplicações - AULA 2

Nos slides seguintes vamos continuar a apresentar as algumas das principais formas de utilização do MatLab. Vamos seguir os apontamentos do **Curso de MatLab de António Anjo**, Ricardo Fernandes e Amaral Carvalho da Universidade de Aveiro. Assim, o nosso plano para esta aula passa por:

- 7) Gráficos - 2D(continuação).
- 8) Gráficos - 3D.
- 9) Introdução a programação usando o MatLab.

# MATLAB - resumo de algumas aplicações

- Exemplo dos comandos anteriores:

```
syms x;
```

```
x = -10 : 0.5 : 10;
```

```
plot(x, x.^3);
```

```
gtext('texto opcional');
```

```
text(3,-3,'texto local');
```

```
title('Grafico de x ao cubo');
```

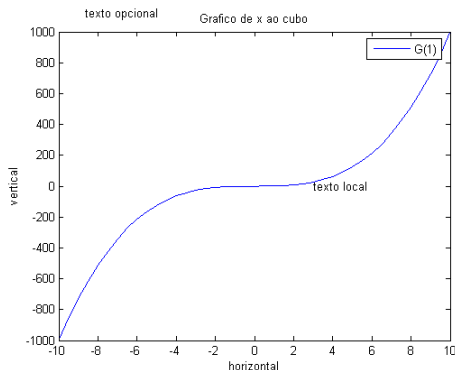
```
xlabel('horizontal');
```

```
ylabel('vertical');
```

```
legend('G1');
```

# MATLAB - resumo de algumas aplicações

- O exemplo anterior produz:



# MATLAB - resumo de algumas aplicações

- **Múltiplos gráficos**

Uma das formas de usar mais do que um gráfico na mesma figura é usar o comando **hold on**. Veja-se o exemplo:

```
x = 0 : pi/40 : 4 * pi
```

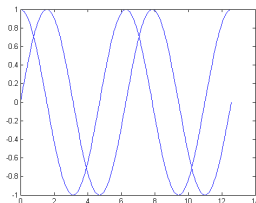
```
plot(x, sin(x))
```

```
x = 0 : pi/40 : 4 * pi
```

```
y = cos(x)
```

```
hold on
```

```
plot(x, y) obtém-se:
```



# MATLAB - resumo de algumas aplicações

- Pode-se alterar o tipo de linha do gráfico, bastando para isso introduzir `'--'` no comando `plot`. Ver mais informações usando **help plot**. Se no exemplo anterior quisermos desenhar a curva  $\sin(x)$  a vermelho e a traço interrompido fazemos:

```
x = 0 : pi/40 : 4 * pi
```

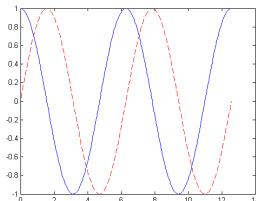
```
plot(x, sin(x), ' -- r')
```

```
x = 0 : pi/40 : 4 * pi
```

```
y = cos(x)
```

```
hold on
```

```
plot(x, y)
```



# MATLAB - resumo de algumas aplicações

- **Múltiplos gráficos no mesmo template**

Usa-se o comando **subplot(m,n,p)**, onde p indica a posição onde se coloca o gráfico. Por exemplo ao escrever o código *subplot(3,3,5)* o MatLab constrói uma matriz  $3 \times 3$  de posições para colocar os gráficos (ver a seguir), e vai permitir desenhar o gráfico na quinta posição,

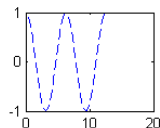
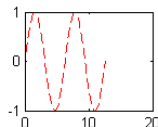
1	2	3
4	<b>5</b>	6
7	8	9

Experimente a escrever:

```
subplot(3,3,5)
x = 0 : pi/40 : 4 * pi
plot(x, sin(x),'- - r')
hold on
subplot(3,3,9)
plot(x, cos(x),'- - b')
```

# MATLAB - resumo de algumas aplicações

- Múltiplos gráficos no mesmo template





# MATLAB - resumo de algumas aplicações

## 8) Gráficos - 3D.

- O comando mais usado para desenhar gráficos 3D é o comando **plot3(x,y,z)**.

Considere o exemplo:

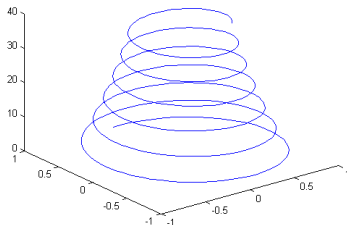
$t = 0 : 0.1 : 40$

$x = \exp(-0.02 * t) .* \sin(t)$

$y = \exp(-0.02 * t) .* \cos(t)$

$z = t$

`plot3(x,y,z)`



# MATLAB - resumo de algumas aplicações

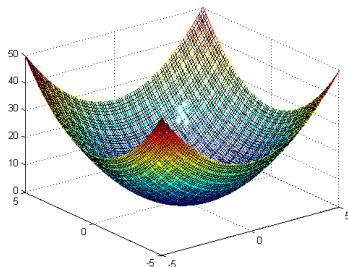
- Outra forma de conseguir desenhar gráficos 3D é com o comando **mesh**.

Considere o exemplo:

```
[x,y] = meshgrid(-5 : 0.1 : 5, -5 : 0.1 : 5)
```

```
z = x.^2+y.^2
```

```
mesh(x,y,z)
```



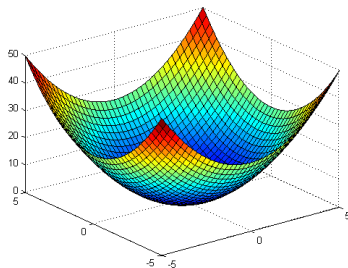
# MATLAB - resumo de algumas aplicações

- Outra forma ainda de conseguir desenhar gráficos 3D é com o comando **surf**.

```
[x,y] = meshgrid(-5 : 0.1 : 5, -5 : 0.1 : 5)
```

```
z = x.^2+y.^2
```

```
surf(x,y,z)
```



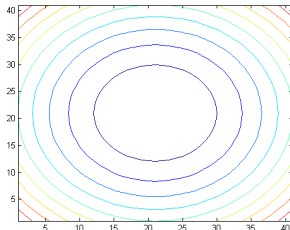
# MATLAB - resumo de algumas aplicações

- "MatLab provides two key commands for plotting surfaces: **mesh** and **surf**. **Mesh** takes the 3D data and creates a wire mesh through adjacent points. On the other hand, **surf** creates a mesh plot with the spaces between the lines, called patches, filled in according to a color scheme based on the z data"

<http://germain.umemat.maine.edu/faculty/bray/Matlab/3dplots.html>

- Para desenhar curvas de nível usa-se o comando **contour**. Experimente a escrever agora a linha de comando

*contour(z)*



# MATLAB - resumo de algumas aplicações

## 9) Introdução a programação usando o MatLab.

### Entrada e saída de dados

O comando **disp(x)** permite escrever no ecrã  $x$ .

- Podemos escrever texto: **disp('texto')**
- Podemos escrever um número  $x$ : **disp(x)**
- Podemos escrever uma frase mais elaborada: **disp('o resultado é:',num2str(y))**

Neste caso temos uma mistura das duas situações anteriores. O comando **num2str** converte o número  $y$  para uma variável do tipo string.

O comando **input** é usado para ler valores introduzidos pelo utilizador.

- Podemos ler valor de  $x$  a seguir a escrever texto:

**x=input('Introduza o valor de x:')**

# MATLAB - resumo de algumas aplicações

## Ciclos

O comando **for(x)** ou **while(x)** é usado para executar um conjunto de instruções um determinado número de vezes.

Escreve-se da seguinte forma:

```
for variavel=expressao  
instrucao 1  
instrucao 2  
end
```

Também pode ser usado o comando **while**, da seguinte forma:

```
while condicao  
instrucao 1  
instrucao 2  
end
```

# MATLAB - resumo de algumas aplicações

## Estruturas de decisão - if/switch

```
if expressao  
instrucao1  
else  
instrucao2  
end
```

Alternativamente podíamos usar:

```
switch expressao  
case valor1  
instrucao 1  
case valor2  
instrucao2  
...  
otherwise  
instrucao3  
end
```

# MATLAB - resumo de algumas aplicações

## Script files

- O Script file é um ficheiro com extensão .m. Este ficheiro pode conter o nosso programa. De seguida o MatLab vai interpretar o código existente no ficheiro e executá-lo. De notar que as variáveis existentes nestes ficheiros são globais, isto é, mesmo depois da execução do programa, as variáveis continuam na memória do computador para serem utilizadas a qualquer momento.
- Depois de gravado o ficheiro com o programa, o utilizador pode correr o mesmo usando a opção **debug+run** ou, na janela de comandos escrever o nome do ficheiro. Nesta segunda opção, o utilizador deve colocar em **current directory** do MatLab, a directoria onde está o ficheiro que quer executar.



# MATLAB - resumo de algumas aplicações

## EXEMPLOS

- EX 1. Defina uma matriz  $A$ , com 5 linhas e 5 colunas. Nessa matriz atribua ao elemento  $(i, j)$  da matriz o valor  $i + j$ . De seguida some todos os valores e calcule a média.
- EX 2. Agora faça o mesmo exercício, mas, lendo os elementos do teclado.
- EX 3. Considere que tem uma função  $\text{custo}(cm, cmo)$ , que se define como:  $\text{custo}(cm, cmo) = \alpha * cm + \beta * cmo$ . Onde  $cm$  é o custo do material e  $cmo$  é o custo de mão de obra.  $\alpha$  e  $\beta$  representam as percentagens dos custos atribuídas a cada uma das variáveis e devem ser introduzidas pelo utilizador. Desenhe o gráfico da função custo.