

# Análise matemática C - PI

Rui Pereira

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

28 Março de 2008

# MATLAB - resumo de algumas aplicações

Nos slides seguintes pretende-se dar algumas luzes de como utilizar o MatLab como ferramenta de ajuda no estudo de funções e representações gráficas. Vamos seguir os apontamentos do **Curso de MatLab de António Anjo**, Ricardo Fernandes e Amaral Carvalho da Universidade de Aveiro. Assim, o nosso plano passa por:

- 1) Efectuar cálculos simples.
- 2) Caracteres especiais.
- 3) Definição de variáveis.
  - 4.1) Algumas funções simples.
  - 4.1) Algumas funções simbólicas.
- 5) Vectores e matrizes.
- 6) Operadores relacionais/lógicos.
- 7) Gráficos - 2D.

# MATLAB - resumo de algumas aplicações

## 1) Efectuar cálculos simples.

- $a + b \rightarrow$  soma a com b;
- $a - b \rightarrow$  subtrai b a a;
- $a * b \rightarrow$  multiplica a por b;
- $a ./ b \rightarrow$  divide a por b;
- $a .^ b \rightarrow$  a elevado a b;
- A regra usada para avaliar as expressões é da esquerda para a direita. A potência é a operação que tem maior precedência seguida da multiplicação/divisão.

# MATLAB - resumo de algumas aplicações

## 2) Caracteres especiais.

- [] → usados para formar vetores e matrizes.
- () → usados para indicar precedências, argumentos de funções, etc.
- . → ponto decimal.
- , → separa elementos de matrizes, argumentos de funções.
- ; → separa linhas de matrizes, no final duma expressão suprime a apresentação de resultados na área de trabalho.
- % → linha de comentário.

# MATLAB - resumo de algumas aplicações

## 3) Definição de variáveis.

O MatLab tem certas regras para a definição de uma variável. Os nomes destas têm um comprimento máximo de 32 caracteres devendo ser iniciados por uma letra, sem espaços nem caracteres de pontuação. Existem algumas variáveis pré definidas:

- **ans** → variável utilizada para resultados.
- **clear** → apaga da memória todas as variáveis definidas pelo utilizador.
- **eps** → precisão relativa de formato double  $2^{-52}$ .
- **i,j** → unidade imaginária.
- **inf** → significa infinito.
- **NaN,nan** → não é um número.

# MATLAB - resumo de algumas aplicações

- **pi** → número  $\pi$ .
- **realmin** → menor número que o computador pode armazenar neste programa.
- **realmax** → maior número que o computador pode armazenar neste programa.
- **who** → lista de variáveis usadas.
- **whos** → lista de variáveis usadas com informações úteis.

# MATLAB - resumo de algumas aplicações

**4.1) Algumas funções simples** O MatLab dispõe de um conjunto de funções matemáticas pré definidas:

- **abs(x)** → Valor absoluto de x.
- **cos(x)** → Coseno de x.
- **sin(x)** → Seno de x.
- **tan(x)** → Tangente de x.
- **cot(x)** → Cotangente de x.
- **acos(x)** → Arco cujo coseno é x.
- **asin(x)** → Arco cujo seno é x.
- **atan(x)** → Arco cuja tangente é x.

# MATLAB - resumo de algumas aplicações

- **sqrt(x)** → Raíz quadrada de x.
- **exp(x)** → Função exponencial de x.
- **rand** → Número aleatório entre 0 e 1.
- **log(x)** → Logaritmo neperiano de x.
- **max(x)** → Função máximo de x.
- **length(x)** → Comprimento de x.
- **size(A)** → Tamanho da matriz A.



# MATLAB - resumo de algumas aplicações

## 4.2) Algumas funções simbólicas

Para representar e manipular variáveis ou funções, que para além de números e variáveis numéricas, contenham incógnitas, estas devem ser declaradas. Por exemplo, se quisermos definir a função  $f(x) = x^2 - 2x + 1$ , uma vez que ela depende de  $x$  devemos declarar da seguinte forma **syms x**.

Para o exemplo apresentado em cima deveríamos então efectuar a sequência de comandos:

```
syms x  
f = x ^ 2 - 2 * x + 1
```

A seguir apresentam-se alguns exemplos de funções que permitem manipular expressões algébricas:

- **compose(f,g)** → função composta  $f(g(x))$

```
g = 2 * x  
compose (f, g)
```

Qual é o resultado esperado? Experimente no MatLab.

# MATLAB - resumo de algumas aplicações

- **simplify(expression)** → Simplifica expressão.
- **solve(expression)** → Dá a solução da expressão.

Exemplo: Resolver  $(x - 5) * (x - 17) = 0$ .

Comando a usar: `solve(' (x - 5) * (x - 17) = 0 ')`

- **roots(p)** → Dá as raízes dum polinómio.
- **diff(f,x)** → Deriva f em ordem a x.
- **int(f,x)** → Integra f em ordem a x.

# MATLAB - resumo de algumas aplicações

## 5) Vectores e matrizes.

### VECTORES

- Um vector pode ser definido de várias maneiras diferentes. A mais simples consiste em especificar as suas componentes.

$$x = [1 \quad 3 \quad 7]$$

Pode também escrever-se:

$$x = [1, 3, 7]$$

Para gerar vectores com valores igualmente espaçados entre um valor inicial e um valor final escreve-se:

$$x = [\textit{valorinicial} : \textit{incremento} : \textit{valorfinal}]$$

Um vector coluna forma-se por exemplo usando o separador ;

$$z = [22; 3; 47].$$

# MATLAB - resumo de algumas aplicações

- Para **retirar** o terceiro elemento de um vector  $x$  fazemos  $x(3)$ .
- Para **retirar o terceiro, o quarto e o quinto elementos** de um vector  $x$  fazemos  $x(3 : 5)$ .
- Para **gerar um vector  $x$  com 7 números aleatórios** fazemos  $x = rand(1, 7)$ .
- Para **retirar o primeiro, o quarto e o sétimo elementos dum vector** fazemos  $x([1, 4, 7]) = []$ . O vector fica assim com menos componentes.
- **Algumas operações com vectores:**
  - $x.*y \rightarrow$  multiplica dois vectores elemento a elemento.
  - $x./y \rightarrow$  divide dois vectores elemento a elemento.
  - $x.^y \rightarrow$  gera vector cujas componentes são  $x$  elevado a  $y$  componente a componente.

# MATLAB - resumo de algumas aplicações

## MATRIZES

- Pode definir-se uma matriz também elemento a elemento fazendo:  
 $A = [1 \ 2 \ 3; 2 \ 3 \ 4; 4 \ 5 \ 6]$  notar que as linhas s ao separadas por ;
- $A = [0 : 0.5 : 2; 1 : 0.25 : 1; 2 : 0.5 : 4]$  gera matriz A com a primeira linha definida com o primeiro elemento 0 e depois elementos igualmente espaçados até 2 com espaçamento de 0.5 e as outras 2 linhas de forma semelhante.
- Para **retirar os elementos da segunda e terceira linhas e primeira e segunda colunas** de uma matriz A fazemos  $A(2 : 3; 1 : 2)$ .
- **Algumas operações com matrizes:**
  - $inv(A) \rightarrow$  dá a inversa da matriz A.
  - $rank(A) \rightarrow$  dá a caracter'ística da matriz A.
  - $det(A) \rightarrow$  dá o determinante da matriz A.
  - $eig(A) \rightarrow$  dá valores próprios da matriz A.
  - $lu(A) \rightarrow$  dá a factorização LU da matriz A.

# MATLAB - resumo de algumas aplicações

## 6) Operadores relacionais/lógicos.

- Os **operadores relacionais** são:  $<$ ,  $\leq$ ,  $=$ ,  $>$ ,  $\geq$ ,  $==$ ,  $\sim=$

Notar que os últimos 2 operadores testam a parte real e imaginária, enquanto os restantes testam apenas a parte real.

- Os **operadores lógicos** são:  $\&$  |  $\mid$   $\sim$

O primeiro é o  $\wedge$  lógico, o segundo é o  $\vee$  lógico, e o terceiro é a negação lógica. De notar que 1 representa o valor verdadeiro e 0 representa o valor falso.

# MATLAB - resumo de algumas aplicações

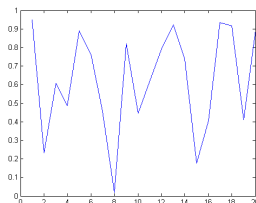
## 7) Gráficos.

- **Gráficos 2D** Para fazer gráficos bidimensionais é habitual usar o comando **plot**. Este comando gera gráficos consoante os parâmetros de entrada. Por exemplo se quiser desenhar um gráfico que represente os valores dum vector gerado aleatoriamente faço por exemplo:

```
x = rand(1,20)
```

```
plot(x)
```

e obtenho:



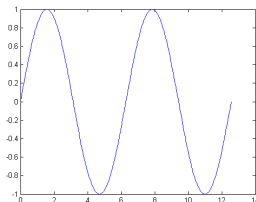
# MATLAB - resumo de algumas aplicações

- O gráfico da função seno entre 0 e  $4\pi$  pode ser gerado como:

$x = 0 : \pi/40 : 4 * \pi$

`plot(x, sin(x))`

e obtenho:





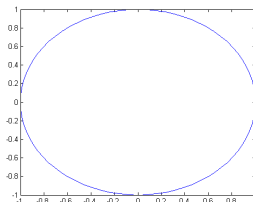
# MATLAB - resumo de algumas aplicações

## EIXOS

- Os eixos no MatLab podem ser reajustados por forma a melhorar a visualização de um gráfico. Para tal usa-se o comando **axis([xmin xmax ymin ymax])**.
- Para voltar aos valores anteriores fazer **axis auto**. Considere-se a circunferência em coordenadas polares  $(x, y) = (\sin(t), \cos(t))$ . Pode-se obtê-la escrevendo:

```
t = 0 : pi/40 : 4 * pi  
plot(sin(t), cos(t))
```

e obtenho:



# MATLAB - resumo de algumas aplicações

- O comando **grid on** torna as grelhas visíveis.
- Introduzir texto no gráfico pode ser conseguido através dos seguintes comandos:

**gtext**('texto') → escreve o texto no gráfico;

**text**(x,y, 'texto') → escreve o texto no gráfico no ponto (x,y);

**title**('texto') → escreve o título texto no gráfico;

**xlabel**('horizontal') → escreve horizontal no eixo do x;

**ylabel**('vertical') → escreve vertical no eixo do y;

**legend**('G1', 'G2', 'G3', ...) → faz a legenda dos diversos gráficos;

# 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('Gráfico de x ao cubo');
```

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

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

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

# MATLAB - resumo de algumas aplicações

- O exemplo anterior produz:

