

## Ficha de Trabalho nº 1

### Matlab: revisão de conceitos básicos

#### 1. Bibliografia

Sebenta e material de apoio disponível no Moodle

Mathworks site: <http://www.mathworks.com/help/techdoc/>

#### 2. Introdução

Neste ficha de trabalho pretende-se que os alunos revejam e relembrem o funcionamento da ferramenta Matlab. Serão abordados os comandos principais, bem como a escrita de algumas funções-

#### 3. Tarefas

##### 3.1 Rever comandos e conceitos básicos do Matlab

Escreva os comandos Matlab que permitam executar as seguintes tarefas:

- Criar uma matriz unidimensional com os números pares até 100
- Criar a transposta da matriz anterior
- Criar a seguinte matriz  
1 2 3  
4 5 6  
7 8 9
- Na matriz anterior, alterar o elemento (1, 3) de 3 para 33.
- Criar uma matriz 3 x 3 e preencher essa matriz com uns.
- Gere uma matriz com dimensão 8x8 constituída por números aleatórios obtidos com a função *randn*. Repita o comando, mas usando a função *rand*. Repare nas diferenças.
- Remova a segunda coluna da matriz anterior.
- Desenho o gráfico da função *seno* para ângulos de 0 a 2pi, com passo de pi/100. Use as funções *sin* e *plot*.
- Usando a função *hold* do matlab acrescente ao gráfico uma linha para o **coseno** dos valores 0 a 2pi. Use a função *cos*.
- Faça as seguintes alterações ao gráfico:
  - Use o comando *legend* para criar uma legenda no gráfico anterior.
  - Use o comando *title* para colocar um título no gráfico anterior.

- c. Altere a cor da linha para verde (use 'g' de green). Use o comando *plot* colocando 'g' no argumento *color*. Faça *help plot* para mais informações.

### 3.2 Funções

(Para comentários use %)

- a) Crie um ficheiro de nome **gera\_matriz.m** e grave-o na sua conta pessoal. Escreva uma função de nome **gera\_matriz** que receba como argumentos dois números inteiros **a** e **b** representando as dimensões da matriz a gerar mais um valor inteiro **x**, correspondendo ao limite máximo dos números aleatórios a colocar na matriz. Use a função **randi** para criar uma matriz **a x b** inicializada com números inteiros aleatórios de 1 a **x**. Teste a função na linha de comando.
- b) Crie um ficheiro de nome **filtra\_matriz.m** e grave-o na sua conta pessoal. Escreva uma função de nome **filtra\_matriz** que receba uma matriz como argumento e que crie um vector unidimensional que contenha os elementos pares da matriz. Para saber o resto da divisão entre dois números use a função **mod**.
- Faça uma implementação utilizando as instruções **for** e **if**
  - Faça outra implementação que a função **find** do Matlab para encontrar uma solução que não utilize as instruções **for** e **if**.
  - Teste a função na linha de comando.
- c) Crie um ficheiro de nome **maior\_num.m** e grave-o na sua conta pessoal. Escreva uma função de nome **maior\_num** que peça vários números ao utilizador. A introdução de dados termina quando o número zero for introduzido. Mostre o maior número introduzido.
- Implemente uma solução sem utilização de vectores/matrizes
  - Implemente outra solução com a utilização de um vector para guardar os valores introduzidos e a função **max** do Matlab para encontrar o maior valor.
- d) Crie um ficheiro de nome **soma\_impares.m** e grave-o na sua conta pessoal. Crie uma função **soma\_impares** que receba uma matriz como argumento e devolva a soma dos elementos ímpares da matriz. Teste a função na linha de comando.
- Faça uma implementação utilizando as instruções **for** e **if**
  - Faça outra implementação que as funções **find** e **sum** do Matlab para encontrar uma solução que não utilize as instruções **for** e **if**.
  - Teste a função na linha de comando
- e) Crie um ficheiro de nome **contapositivos\_matriz.m** e grave-o na sua conta pessoal. Escreva uma função **contapositivos\_matriz** que receba um vector como argumento e imprima no monitor quantos números positivos se encontram nesse vector.
- Faça uma implementação utilizando as instruções **for** e **if**
  - Faça outra implementação que as funções **find** e **numel** do Matlab para encontrar uma solução que não utilize as instruções **for** e **if**.
  - Teste a função na linha de comando

- f) Crie um ficheiro de nome **maiores\_matriz.m** e grave-o na sua conta pessoal. Escreva função **maiores\_matriz** que receba um vector como argumento e devolva o maior número do vector bem como a posição (índice) onde esse número se encontra. Teste a função na linha de comando.
- Faça uma implementação utilizando as instruções **for** e **if**
  - Faça outra implementação que as funções **find** e **max** do Matlab para encontrar uma solução que não utilize as instruções **for** e **if**.
  - Teste a função na linha de comando