

INTRODUÇÃO AO MATLAB

Murilo F. Tomé
ICMC-USP

- Funções
- Solução sistemas lineares
- Arquivos



Arquivos FUNÇÕES

- Um arquivo-M com a palavra “function” no início da 1a. linha é interpretado como **arquivo função**.
- Difere de um **script** pelos argumentos que são passados.
- Variáveis definidas são locais à função (não podem ser operadas globalmente no espaço de trabalho).



Arquivos FUNÇÕES

Forma geral:

`function` [$p_{s_1}, p_{s_2}, \dots, p_{s_m}$] = **NOME** ($p_{e_1}, p_{e_2}, \dots, p_{e_n}$)
comandos

$p_{s_1}, p_{s_2}, \dots, p_{s_m}$ - Parâmetros de saída;

$p_{e_1}, p_{e_2}, \dots, p_{e_n}$ - Parâmetros de entrada;



Chamada de função

≫ $[as_1, as_2, \dots, as_m] = \mathbf{NOME}(ae_1, ae_2, \dots, ae_n)$

Os argumentos $(ae_1, ae_2, \dots, ae_n)$ são passados (copiados) aos parâmetros de entrada $(pe_1, pe_2, \dots, pe_n)$.

Dentro da função, são realizados cálculos que são passados aos argumentos de saída $[as_1, as_2, \dots, as_m]$.



Exemplo - função

```
function [area, perimetro] = CIRCULO(RAIO)
```

```
area = pi*RAIO*RAIO;
```

```
perimetro = 2.0*pi*RAIO;
```

```
>> r = 3.0;
```

```
>> [x1,x2] = CIRCULO(r)
```

```
x1 = 28.2743          x2 = 18.8496
```

```
>> s = 4.0;
```

```
>> [y1,y2] = CIRCULO(s)
```

```
y1 = 50.2655          y2 = 25.1327
```



Solução sistemas lineares pelo MATLAB

Dado o sistema linear

$$\mathbf{Ax} = \mathbf{b}. \text{ Ex: } \begin{bmatrix} 7 & 1 & 2 & 3 \\ 1 & 7 & 2 & 3 \\ 1 & 2 & 7 & 3 \\ 1 & 2 & 3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

» det(A)

ans = 1560 (A é não-singular)

Como encontrar a solução pelo MATLAB?



Uso explícito da inversa de A

$$\mathbf{x} = A^{-1}b$$

Para o sistema linear dados temos:

$$\gg \mathbf{x} = \text{inv}(A)*b$$

$$\gg A*\mathbf{x}$$

x =

-0.1346

0.0321

0.2321

0.4821

ans =

1.0000

2.0000

3.0000

4.0000



Decomposição LU

Pode também utilizar a decomposição LU

No MATLAB, essa operação é representada pelo operador **divisão a esquerda** (\backslash):

```
» x = A\b
```

```
x =
```

```
-0.1346
```

```
0.0321
```

```
0.2321
```

```
0.4821
```



Arquivos

Comando fopen: Esse comando é utilizado para abrir um arquivo em MATLAB.

Forma geral: `fopen('nome arquivo', 'permissão')`

'permissão' pode ser: `'r'` (read) ou `'w'` (write)



Exemplo - abre e cria arquivo

```
>> x = -1:0.2:1
```

```
>> y = [x; exp(x)]
```

```
>> fid = fopen('exp.dat','w')
```

```
>> fprintf(fid,'%6.4f %12.6f\n',y)
```

```
>> fprintf(fid,'%6.4f %12.6f\n',x,exp(x)) outra maneira
```

```
>> fprintf(fid,'%f %f\n',x,exp(x)) outra maneira de imprimir
```



Comando fscanf e fclose

Comando utilizado para ler arquivos

Forma geral: `fscanf(FID, '%f' ou '%d', N)`
`fscanf(FID, '%f' ou '%d', [M,N])`

Se não houver `N` ou `[M,N]`, o arquivo é lido até o fim.

Comando `fclose`: esse comando é utilizado para fechar um arquivo aberto.

Forma geral: `fclose(nome-arquivo)`



Exemplo - abre e imprime arquivo

- » `fid = fopen('expa.dat','r');` abre o arquivo `expa.dat`
- » `[Y, N] = fscanf(fid, '%f %f\n');`
dados lidos são armazenados na matriz `Y`
- » `fclose(fid);` fecha o arquivo
- » `arqui2 = fopen('teste.dat', 'w');` abre `teste.dat` que será criado
- » `fprintf(arqui2, '%f %f\n', Y);` escreve (cria) no arquivo
- » `fclose(arqui2);` fecha o arquivo `teste.dat`



Arquivos expa.dat e teste.dat

-1.0000	0.367879	-1.000000	0.367879
-0.8000	0.449329	-0.800000	0.449329
-0.6000	0.548812	-0.600000	0.548812
-0.4000	0.670320	-0.400000	0.670320
-0.2000	0.818731	-0.200000	0.818731
0.0000	1.000000	0.000000	1.000000
0.2000	1.221403	0.200000	1.221403
0.4000	1.491825	0.400000	1.491825
0.6000	1.822119	0.600000	1.822119
0.8000	2.225541	0.800000	2.225541
1.0000	2.718282	1.000000	2.718282

