

CÁLCULO NUMÉRICO / 1º semestre -2003

AULA 1: Principais comandos do MATLAB

Operações aritméticas básicas:

+ adição
- subtração
* multiplicação
/ divisão
^ potenciação

Comandos importantes:

help, who, whos,

1. Determinação de zeros de função

- **fzero** : encontra zero de função contínua de uma variável.

`x = fzero(fun,x0)`

`x = fzero(fun,x0,options)`

`x = fzero(fun,x0,options,P1,P2,...)`

`[x,fval] = fzero(...)` => retorna o valor da função no ponto x

Note que fun é um arquivo **M-file** que define a função e deve ser salvo com **fun.m**

- **Exemplo 1**

Encontrar o zero da função $f(x) = x^3 - 2x - 5$ próximo a 2.

```
>> x = fzero(inline('x.^3-2*x-5'), 2)
```

x =

2.0946

Outra forma de resolver...

Escrever um arquivo M-file chamado f.m. O símbolo @ indica que f é uma função. Para abrir arquivo abrir o Matlab e ir em (file => New => M-file).

```
function y = f(x)
```

```
y = x.^3-2*x-5;
```

```
>>z = fzero(@f, 2)
```

```
z =  
    2.0946
```

- **Exemplo 2**

Encontrar o zero da função cosseno entre 1 e 2

```
>>x = fzero(@cos,[1 2])
```

```
x =  
    1.5708
```

2. Definição de vetores e matrizes

- **Algumas formas de se definir um vetor...**

```
>> x=[1 2 3 4 5 6]  
x =
```

```
    1    2    3    4    5    6
```

```
>> x=(1:6)
```

```
x =
```

```
    1    2    3    4    5    6
```

```
>> x(2:2:10)
```

```
ans =
```

```
    2    4    6    8   10
```

```
>> x=(1:2:20)
```

```
x =
```

```
1  3  5  7  9 11 13 15 17 19
```

```
>> linspace(1,10,10)
```

```
ans =
```

```
1  2  3  4  5  6  7  8  9 10
```

```
>> linspace(1,10,5)
```

```
ans =
```

```
1.0000  3.2500  5.5000  7.7500 10.0000
```

O comando `linspace (x1, x2, N)` gera N números igualmente espaçados entre os pontos `x1` e `x2`.

- Operações com vetores

- **Adição/Subtração de vetores**

```
>> x=[1 2 3 4 5 6]
```

```
x =
```

```
1  2  3  4  5  6
```

```
>> x=(1:6)
```

```
x =
```

```
1  2  3  4  5  6
```

```
>> y=(3:8)
```

```
y =
```

```
3  4  5  6  7  8
```

```
>> z=[x y]  (concatena dois vetores)
```

z =

1 2 3 4 5 6 3 4 5 6 7 8

>> x+y

ans =

4 6 8 10 12 14

>> x-y

ans =

-2 -2 -2 -2 -2 -2

>> x-3 (subtração por escalar)

ans =

-2 -1 0 1 2 3

- Multiplicação/Divisão de vetores

>> x=(1:6)

x =

1 2 3 4 5 6

>> y=(5:10)

y =

5 6 7 8 9 10

>> x.*y (multiplicação elemento por elemento)

ans =

5 12 21 32 45 60

>> 5*x-2 (multiplicação e subtração por escalar)

ans =

```
3 8 13 18 23 28
```

```
>> x./y
```

```
ans =
```

```
0.2000 0.3333 0.4286 0.5000 0.5556 0.6000
```

Atenção: a divisão sem o ponto é *divisão matricial*!!!

```
>> x=(1:6) (potenciação)
```

```
x =
```

```
1 2 3 4 5 6
```

```
>> x.^2 (realiza a potenciação em cada elemento do vetor)
```

```
ans =
```

```
1 4 9 16 25 36
```

```
>> ans' (resultado transposto)
```

```
ans =
```

```
1
4
9
16
25
36
```

Para transpor um vetor ou uma matriz basta colocar o símbolo '.

- **Como definir uma matriz:**

```
>> A=[1 3 6 7; 8 0 5 2; 1 0 9 6]
```

```
A =
```

```
1 3 6 7
8 0 5 2
1 0 9 6
```

- Operações com Matrizes

```
>> B=A(:) (transformando matriz em vetor coluna)
```

```
B =
```

```
1
8
1
3
0
0
6
5
9
7
2
6
```

```
>> A(1,1)=1000 (alterando apenas um elemento da matriz)
```

```
A =
```

```
1000    3    6    7
      8    0    5    2
      1    0    9    6
```

```
>> C= [1 3 6;2 0 5; 9 8 1]
```

```
>> Z=[A C] (concatenando as matrizes A e C)
```

```
Z =
```

```
1000    3    6    7    1    3    6
      8    0    5    2    2    0    5
      1    0    9    6    9    8    1
```

```
>> 8*C (multiplicação por escalar)
```

```
ans =
```

```
8 24 48
16 0 40
```

```
72 64 8
```

```
>> A' (matriz transposta)
```

```
ans =
```

```
1000      8      1
      3      0      0
      6      5      9
      7      2      6
```

```
>> det(C) (verificando condição de existência da inversa)
```

```
ans =
```

```
185 (existe inversa pois o determinante é diferente de zero)
```

```
>> inv(C) (devolve a inversa da matriz quadrada)
```

```
ans =
```

```
-0.2162  0.2432  0.0811
 0.2324 -0.2865  0.0378
 0.0865  0.1027 -0.0324
```

Obs. As operações de soma, subtração, multiplicação e divisão são análogas às operações vistas com um vetor. Deve-se apenas verificar as dimensões das matrizes (vetores) antes através dos comandos *length* e *size*.

```
>> x=(1:5)
```

```
x =
```

```
1 2 3 4 5
```

```
>> length(x) (devolve o comprimento do vetor)
```

```
ans =
```

```
5
```

```
>> B=[1:5;2:6;0:4] (definindo uma matriz)
```

```
B =
```

```
1 2 3 4 5
```

```

2 3 4 5 6
0 1 2 3 4

```

```
>> size(B)    ( devolve o número de linhas e colunas da matriz)
```

```
ans =
```

```
3 5
```

Outros comandos associados a matrizes: (ver help matlab!!!)

eye (matriz identidade), eig (autovalores e autovetores matriz quadrada), ones, zeros, norm, cond, diag, reshape ...

3. Resolução de sistemas lineares

Considere o seguinte sistema linear: ($Ax=b$)

$$3x_1 + 2x_2 + 4x_3 = 1$$

$$1x_1 + 1x_2 + 2x_3 = 2$$

$$4x_1 + 3x_2 + 2x_3 = 3$$

Definindo a matriz A e o vetor b no matlab...

```
>> A=[3 2 4; 1 1 2; 4 3 2]    (matriz coeficientes lado esquerdo)
```

```
>> b=[1; 2; 3]    (vetor coluna lado direito)
```

```
>> x=inv(A)*b    (uso direto da matriz inversa se det(A) ≠ 0!)
```

```
x =
```

```

-3
5
0

```

ou

```
>> x=A\b    (utiliza técnica Eliminação de Gauss)
```

```
x =
```

```

-3
5

```

0

Obs. Vale lembrar que quando o número de equações for maior que o número de incógnitas (caso superdeterminado) a solução é encontrada através da minimização do erro quadrado em $Ax - b = 0$ (*solução dos mínimos quadrados*).

- Fatorização LU

```
>> [l,u,p]=lu(A) (matriz deve ser quadrada!)
```

l =

```
0.7500 -1.0000 1.0000
0.2500 1.0000 0
1.0000 0 0 (matriz triangular inferior)
```

u =

```
4.0000 3.0000 2.0000
0 0.2500 1.5000
0 0 4.0000 (matriz triangular superior)
```

p =

```
0 0 1
0 1 0
1 0 0 (matriz de permutação)
```

4. Formas de plotar gráficos

- **Exemplo 1: (usando comando *plot*)**

```
>> x=(1:6)
```

x =

```
1 2 3 4 5 6
```

```
>> y=3*x.^2+1
```

y =

```
4 13 28 49 76 109
```

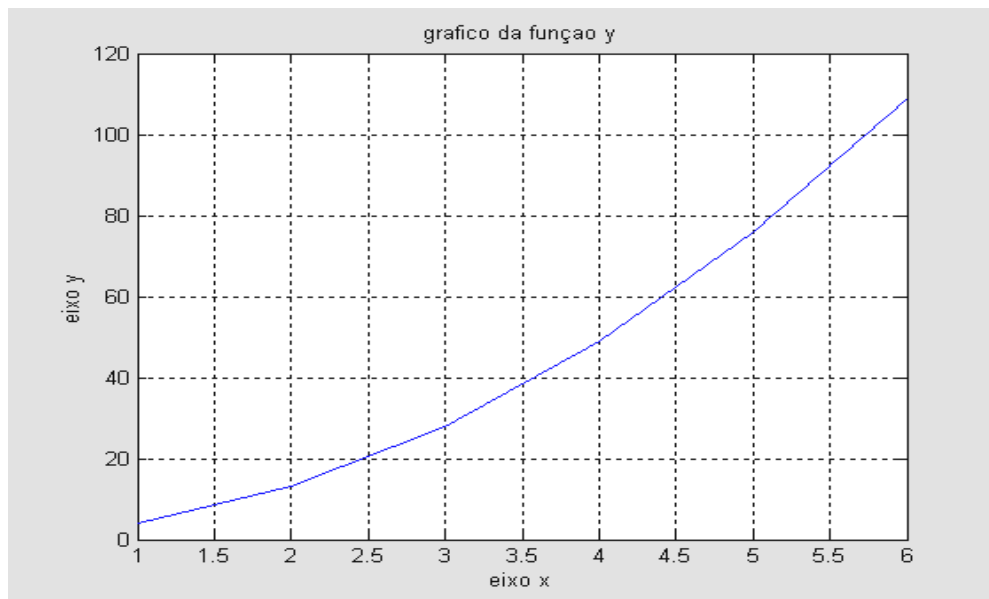
```
>> plot(x,y)

>> title('grafico da função y')

>> xlabel('eixo x')

>> ylabel('eixo y')

>> grid
```



- **Exemplo 2: (usando comando *fplot*)**

Definindo variáveis simbólicas:

```
>> syms x

>> y='x^3+4*x+1'

y =

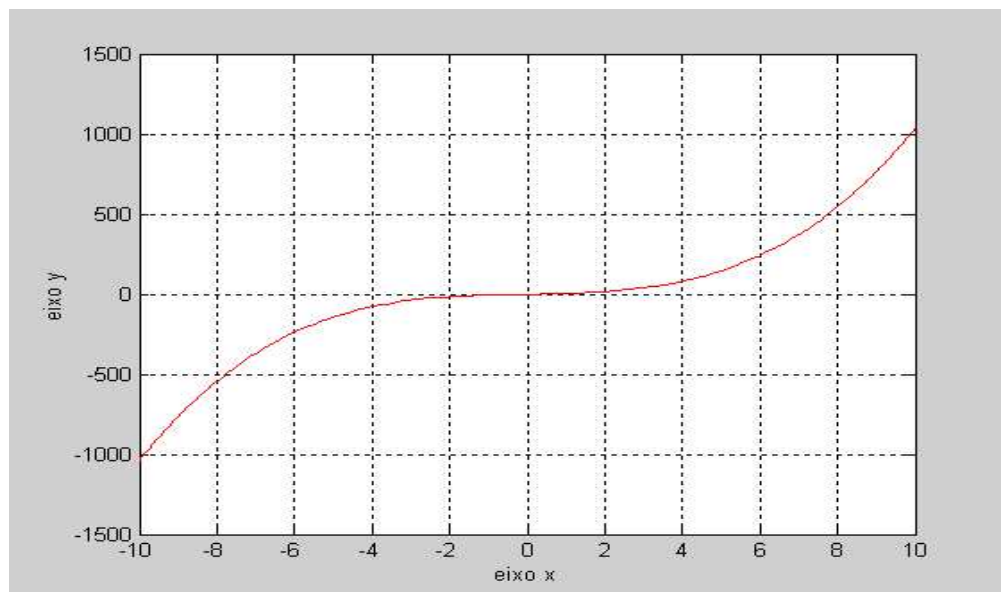
x^3+4*x+1

>> fplot(y,[-10 10],'r')    ( plota gráfico em vermelho no intervalo [-10 10] )

>> xlabel('eixo x')

>> ylabel('eixo y')
```

```
>> grid
```



```
>> syms x
```

```
>>
```

```
>> y='sin(x)'
```

```
y =
```

```
sin(x)
```

```
>> fplot(y,[0 2*pi],'r')
```

```
>> hold on (congela a figura)
```

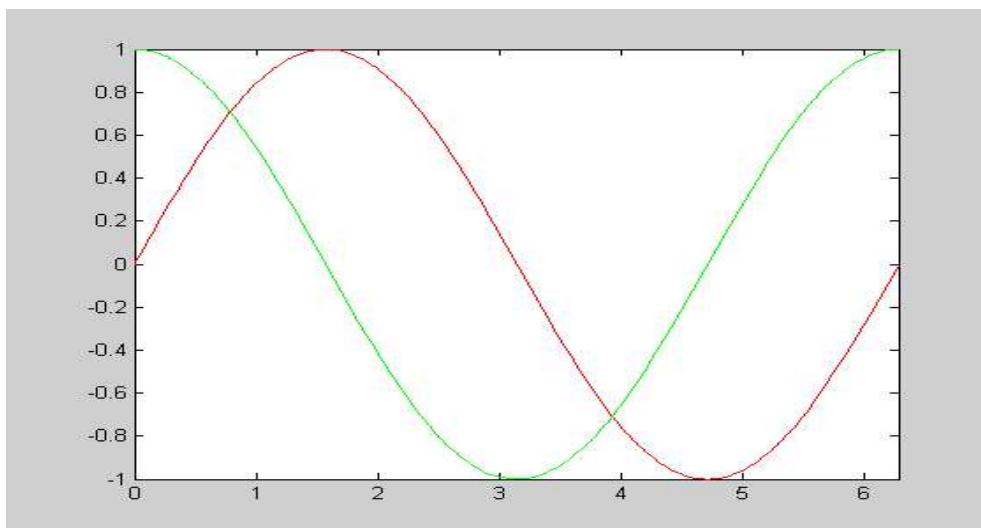
Obs. Para descongelar use o comando hold off.

```
>> z='cos(x)'
```

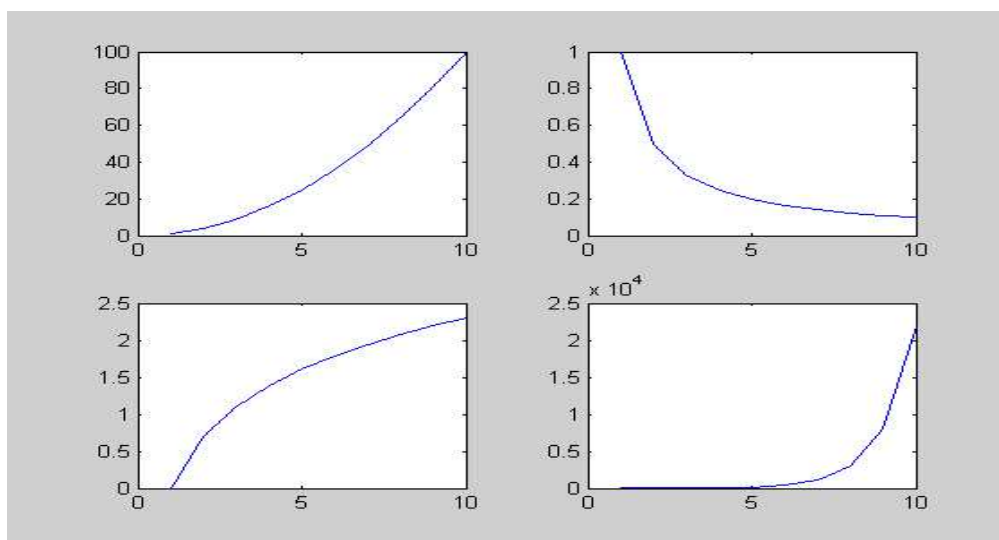
```
z =
```

```
cos(x)
```

```
>> fplot(z,[0 2*pi],'g')
```



```
>> x=(1:10)
>> subplot(2,2,1)
>> plot(x.^2)
>> subplot(2,2,2)
>> plot(1./x)
>> subplot(2,2,3)
>> plot(log(x))
>> subplot(2,2,4)
>> plot(exp(x))
```



Outros comandos associados a gráficos: (ver help matlab!!)

Subplot, hold, figure, axis, legend...