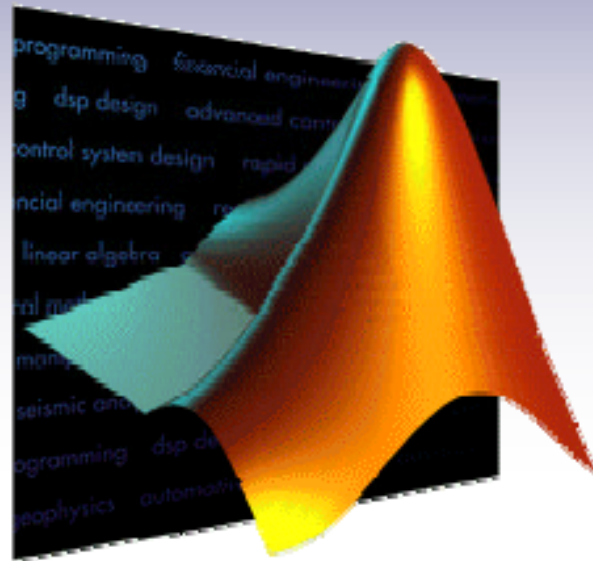


Curso MATLAB 6

Introdução à Solução de Problemas de Engenharia



The
MATH
WORKS
Inc.

Instrutor: Marcelo Escobar
Complementação da Introdução

Complementação da Introdução:

Gráficos

Análise de Dados

Polinômios

Ajuste de Curvas

Números Complexos

Funções Temporais

Gráficos:

```
>>help graph2d  
>>help graph3d  
>>help specgraph
```

```
>>help plot  
>>help fplot [alternativa]  
>>x=1:0.1:3;  
>>y=sin(x);  
>>plot(x,y) [plotagem y(ordenada) versus x(abscissa)]  
>>fplot('sin(x)',[0 3]) [plota entre os limites]
```

```
>>grid on [exibe grid]  
>>xlabel('x'); [ texto no eixo das abscissas]  
>>ylabel('y'); [ texto no eixo das ordenadas]  
>>title(' Grafico do seno(x)') [titulo do grafico]  
>>text [ exibe texto em ponto (x,y) indicado]  
>>legend(' seno'); [exibe legenda]
```

Gráficos:

Plotagem de dois gráficos no mesmo eixo:

Opção 1: Plotar usando o mesmo comando

```
>>x=linspace(0,3);  
>>y=sin(x);z=cos(x);  
  
>>plot(x,y,x,z) [plota os dois no mesmo eixo]
```

Opção 2: Plotar um de cada vez

```
>>x=linspace(0,3);  
>>y=sin(x);z=cos(x);  
  
>>plot(x,y)  
>>hold on [ segura o grafico atual]  
>>plot(x,z)
```

Gráficos:

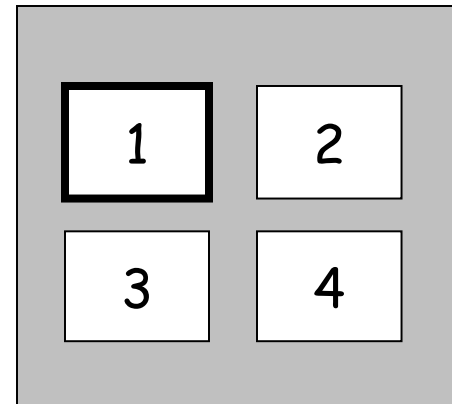
Plotagem de dois gráficos na mesma janela em eixos diferentes:

```
>>x=linspace(0,3);  
>>y=sin(x);z=cos(x);
```

```
>>help subplot
```

exemplo: subplot(2,2,1)

```
>>subplot(2,1,1)  
>>plot(x,y)  
>>subplot(2,1,2)  
>>plot(x,z)
```



Subplot(a,b,c)
a= numero de eixos na horizontal
b=número de eixos na vertical
c=posição do gráfico

Gráficos:

Propriedades dos Gráficos:

Gráficos, janelas, figuras, eixos,..., são objetos para o matlab. Todo objeto possui um identificador chamado handle.

Com o identificador do objeto ,h,podemos:

>>get(h) [obtem todas as propriedades atuais]

>>set(h) [propriedades possíveis de serem ajustadas]

Como obter o handle do Objeto:

Veremos mais detalhes no tópico de interfaces.

Por enquanto, se pedirmos para plotar fornecendo um retorno para o comando plot, esse retorno será o handle.

>>h=plot(x,y) [h é o handle do gráfico]

Gráficos:

>>set(h,'linewidth',2) [espessura da linha]

>>set(handle, 'prop1', valor1,'prop2',valor2)

As propriedades color e linestyle podem ser fornecidas no comando plot logo após a ordenada:

>>plot(x,y, 'r') [plota o gráfico em vermelho]

g-green y-amarelo c- cyan r-red
b-blue m=magenta

Estilo de linha:

-continua , -- tracejada, o-bolinhas ,> setas

*asterisco, p-estrela, h-estrela

>>plot(x,y,'o') [plota os pontos como bolinhas]

Ou ainda: >>plot(x,y,'r-') [xy em linha continua e vermelha]

Gráficos:

Todas as propriedades do grafico , podem ser fornecidas no próprio comando plot:

```
>>plot(x,y,'m- -', 'linewidth',2)
```

[plota xy em cor magenta com linha tracejada com espessura 2]

A janela do matlab é um objeto também e possui identificador 0:

```
>>set(0,'DefaultBackgroundColor',[1 1 1])
```

[coloca a cor branca no background do grafico ao invés de cinza]

Obs: As cores podem ser fornecidas por string, ex: 'r' -red
Ou por um vetor de cores na forma RGB(red-green-blue).Os valores de cada posição variam de 0 a 1 :

Color=[1 1 1] branco

[1 0 0] vermelho

[0 1 0] verde

Color=[0 0 0] preto

[0 0 1] azul

[0.8 0.8 0.8] cinza

Gráficos:

Escalas não lineares:

```
>>help loglog    [escala log log]  
>>help semilogx [escala log em x]  
>>help semilogy [escala log em y]
```

Gráficos Especiais:

```
>>help polar [ coordenadas polares]  
>>help hist  [histograma]  
>>help stairs [linha degrau]  
>>help stem , stem3 [sequência discreta]  
>>help bar, bar3    [gráficos de barra 2D,3D]  
>>help pie, pie3     [gráficos de torta 2D,3D]
```

```
>>help specgraph
```

Gráficos 3D:

>>help plot3d

>>help meshgrid [cria um grid,uma rede de pontos para a plotagem de uma superfície]

>>mesh [plota superfície]

>>[X,Y] = meshgrid(-2:.2:2, -2:.2:2);

>>Z = X .* exp(-X.^2 - Y.^2);

>>mesh(Z) [plota a superfície]

>>surf(Z) [plota a superfície colorindo a rede]

>>surfc(Z) [plota a superfície colorindo a rede com curvas de nível]

>>contour(Z) [traça as curvas de nível]

>>help graph3d

Análise de Dados:

>>help datafun

>>help mean [média de um vetor]

>>help min [mínimo de um vetor]

>>help max [máximo de um vetor]

>>help sort [coloca em ordem crescente]

>>help std [desvio padrão]

>>help var [variância]

Dica: se quisermos ordem decrescente,

>>x=[3 1 2]

>>x1=sort(x) [x1=[1 2 3]]

>>x2=x1(end:-1:1); [x2=[3 2 1]]

>>help stats [toolbox de estatística]

Polinômios:

>>help polyfun

Como criar?

```
>>p=[1 -2 0];    [ x2 +2x]  
>>q=[2 0];      [2x]
```

Manipulação:

Soma: +	Subtração: -
Produto: conv(p,q)	Divisão: dconv(p,q)

Raízes: roots(p)

```
>>poly(raizes); [ cria o polinômio a partir das raízes]  
>>polyval(p,2); [ avalia o polinômio para um dado valor]
```

Polinômios:

Integral de polinômios:

```
>>help polyint
```

Derivada de polinômios:

```
>>help polyder
```

Ajuste de Polinômio a um conjunto de pontos:

```
>>help polyfit
```

```
>>polyfit(x,y,n) [ ajusta a x,y um polinômio de grau n]
```

```
>>taylortool [ ajuste usando série de Taylor]
```

Interpolação:

```
>>help interp1
```

```
>>interp1(x,y,vx) [ interpola para dados x,y um valor vx]
```

```
>>help interp2 [interpolação em duas dimensões]
```

```
>>help interp3 [interpolação em três dimensões]
```

Splines:

Splines são polinômios por partes, utilizado na interpolação. Isso evita o mau comportamento na interpolação com polinômios de ordem elevada.

```
>>help splines [splines toolbox]
```

Interpolação com Splines:

```
>>help spline
```

```
>>help pchip [Polinômios Cúbicos de Hermite]
```

```
>>spline(x,y,xx) [retorna yy interpolado]
```

Ou:

```
>>pp=spline(x,y) [cria a estrutura spline]
```

```
>>yy=ppval(pp,xx); [retorna yy interpolado]
```

Ajuste de Curvas:

Podemos querer ajustar a um conjunto de dados x, y , uma curva que não seja polinomial. Nesse caso, podemos usar o comando do toolbox de otimização `lsqcurvefit`

```
>>help lsqcurvefit
```

Devemos criar um arquivo que contem o ajuste desejado.

```
function F = myfun(x,xdata)
F = x(1)*sin(xdata)+x(2);
```

```
>>xdata=[5;4;6];
>>ydata=[ 3.1232; 3.7296;5.1618]
>>x = lsqcurvefit('myfun', [2,7], xdata, ydata)
      ↘ chute inicial
```

Solução:

```
x=[3;6]           F=3sin(x)+6
```

Números Complexos:

```
>> A= 2+ 3*i ou >>A= 2+ 3*j
```

Se as variáveis i e j não existem no workspace, o matlab assume o valor $\sqrt{-1}$ para elas.

A manipulação é a mesma feita para números reais, além de alguns comandos específicos.

```
>>iscomplex [função relacional]  
>>help real [parte real]  
>>help imag [parte imaginaria]  
>>abs(A) [ modulo]  
>>angle(A) [ângulo]
```


Funções Temporais:

>>help timefun

>>date

>>clock

>>calendar

Os comandos abaixo, servem para retornar o tempo necessário para a execução de comandos:

>>tic [inicializa a contagem]

>>toc [finaliza a contagem]

>>etime

Útil para a comparação da eficiência de métodos.