Notas de Aula
MatLab - 4

Routo Terada

www.ime.usp.br/~rt

Depto. C. da Computação - USP

Bibliografia:

E. Y. Matsumoto, MatLab6 Fundamentos de Programação, Edit. Érica, 2000

K. Chen et al., Mathematical explorations with MatLab, Cambridge University Press 1999

D. Hanselman et al., MatLab 5 -- Guia do Usuário, Editora Makron 1999

Como compilar uma função

```
function [Mprod] = teste_mcc(A,B)
Mprod=A*B;  % calcula produto das matrizes A e B
```

(1) M-file teste_mcc.m editado e salvo no diretório, por ex. c:\

(2) mudar Current Directory para c:\

mcc -x teste_mcc

teste_mcc.c e teste_mcc.h

(3) Arquivos gerados no diretório c:\

Parte do arquivo teste_mcc.c gerado pelo mcc

```
/*
 * The function "Mteste_mcc" is the implementation version of the
"teste_mcc"
 * M-function from file "C:\MATLABR12\work\teste_mcc.m" (lines 1-2). It
 * contains the actual compiled code for that M-function. It is a static
 * function and must only be called from one of the interface functions,
 * appearing below.
 */
/*
 * function [Mprod] = teste_mcc(A,B)
 */
static mxArray * Mteste_mcc(int nargout_, mxArray * A, mxArray * B) {
    mexLocalFunctionTable save_local_function_table_ =
mclSetCurrentLocalFunctionTable(

&_local_function_table_teste_mcc);
    mxArray * Mprod = mclGetUninitializedArray();
    mclCopyArray(&A);
    mclCopyArray(&B);
    /*
     * Mprod=A*B;  % calcula produto das matrizes A e B
     */
    mlfAssign(&Mprod, mclMtimes(mclVa(A, "A"), mclVa(B, "B")));
    mclValidateOutput(Mprod, 1, nargout_, "Mprod", "teste_mcc");
    mxDestroyArray(B);
    mxDestroyArray(A);
    mclSetCurrentLocalFunctionTable(save_local_function_table_);
    return Mprod;
}
```

```
function [p,S,mu] = polyfit(x,y,n)
%POLYFIT Fit polynomial to data.
%   POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X) of
%   degree N that fits the data, P(X(I))~=Y(I), in a least-squares sense.
%
%   [P,S] = POLYFIT(X,Y,N) returns the polynomial coefficients P and a
%   structure S for use with POLYVAL to obtain error estimates on
%   predictions.  If the errors in the data, Y, are independent normal
%   with constant variance, POLYVAL will produce error bounds which
%   contain at least 50% of the predictions.
%
%   Copyright 1984-2000 The MathWorks, Inc.
%   $Revision: 5.14 $  $Date: 2000/06/13 18:20:09 $

% The regression problem is formulated in matrix format as:
%
%    y = V*p    or
%
%          3  2
%    y = [x  x  x  1] [p3
%                      p2
%                      p1
%                      p0]
%
% where the vector p contains the coefficients to be found.  For a
% 7th order polynomial, matrix V would be:
%
% V = [x.^7 x.^6 x.^5 x.^4 x.^3 x.^2 x ones(size(x))];

if ~isequal(size(x),size(y))
   error('X and Y vectors must be the same size.')
end

x = x(:);
y = y(:);

if nargout > 2
   mu = [mean(x); std(x)];
   x = (x - mu(1))/mu(2);
end
```

(1) M-file polyfit.m editado e salvo no diretório, por ex. c:

```
% Construct Vandermonde matrix.
V(:,n+1) = ones(length(x),1);
for j = n:-1:1
  V(:,j) = x.*V(:,j+1);
end

% Solve least squares problem, and save the Cholesky factor.
[Q,R] = qr(V,0);
ws = warning('off');
p = R\(Q'*y);    % Same as p = V\y;
warning(ws);
if size(R,2) > size(R,1)
  warning('Polynomial is not unique; degree >= number of data points.')
elseif condest(R) > 1.0e10
   if nargout > 2
      warning(sprintf( ...
         ['Polynomial is badly conditioned. Remove repeated data points.']))
   else
      warning(sprintf( ...
         ['Polynomial is badly conditioned. Remove repeated data points\n' ...
           '        or try centering and scaling as described in HELP POLYFIT.']))
   end
end
r = y - V*p;
p = p.';          % Polynomial coefficients are row vectors by convention.

% S is a structure containing three elements: the Cholesky factor of the
% Vandermonde matrix, the degrees of freedom and the norm of the residuals.

S.R = R;
S.df = length(y) - (n+1);
S.normr = norm(r);
```

(1) M-file polyfit.m editado e salvo no diretório, por ex. c:

(2) mudar Current Directory para c:

```
>> mcc -x polyfit
Please choose your compiler for building external interface (MEX) files:


Select a compiler:
[1] Lcc C version 2.4 in E:\MATLABR12\sys\lcc
[2] Microsoft Visual C/C++ version 6.0 in c:\VisualStudio6

[0] None

Compiler: 2

Please verify your choices:

Compiler: Microsoft Visual C/C++ 6.0
Location: c:\VisualStudio6

Are these correct?([y]/n): y
```

(3) Gerou no diret c:\ os arquivos polyfit.c polyfit.h polyfit.dll

```
/*
 *
 * % Construct Vandermonde matrix.
 * V(:,n+1) = ones(length(x),1);
 */
mclArrayAssign2(
 &V,
 mlfOnes(mlfScalar(mclLengthInt(mclVa(x, "x"))), _mxarray6_, NULL),
 mlfCreateColonIndex(),
 mclPlus(mclVa(n, "n"), _mxarray6_));
/*
 * for j = n:-1:1
 */
{
    mclForLoopIterator viter__;
    for (mclForStart(&viter__, mclVa(n, "n"), _mxarray7_, _mxarray6_);
         mclForNext(&viter__, &j);
         ) {
        /*
         * V(:,j) = x.*V(:,j+1);
         */
        mclArrayAssign2(
          &V,
          mclTimes(
            mclVa(x, "x"),
            mclVe(
              mclArrayRef2(
                mclVsv(V, "V"),
                mlfCreateColonIndex(),
                mclPlus(mclVv(j, "j"), _mxarray6_)))),
          mlfCreateColonIndex(),
          mclVsv(j, "j"));
    /*
     * end
     */
    }
    mclDestroyForLoopIterator(viter__);
}
/*
 *
 * % Solve least squares problem, and save the Cholesky factor.
 * [Q,R] = qr(V,0);
 */
```

Parte do arquivo polyfit.c gerado pelo mcc

Solução de equações: solve()

```
>>x=sym('x'); % declara x como variavel simbólica
>>sol=solve('x^2+4*x+3=0', x)  % resolver equação
```

```
sol = [ -3] [ -1]
```

```
>> a=sym('a');
>> sol=solve('x^2-a*x-1=0', x)  % resolver em função de a
```

```
sol = [ 1/2*a+1/2*(a^2+4)^(1/2)]
      [ 1/2*a-1/2*(a^2+4)^(1/2)]
```

Solução de equações: solve()

```
>>a=sym('a');
>>sol=solve('x^2-a*x-1=0', a)
```

```
sol =  (x^2-1)/x
```

```
>> sol=roots([1 -1 3 2 1 -1]) % resolve x^5-x^4+3x^3+2x^2+x-1
```

```
sol =
   0.8203 + 1.7971i
   0.8203 - 1.7971i
  -0.5331 + 0.5639i
  -0.5331 - 0.5639i
   0.4255
```

Solução de equações: solve()

EXERCÍCIO

Resolver: $$6x^3+11x+37.8=0$$

Resp.:

[-1.5201740189901002825247469669899]

[.76008700949505014126237348349496 - 1.8885259117477627725734385669866 i]

[.76008700949505014126237348349496 + 1.8885259117477627725734385669866 i]

Solução de equações: solve()

EXERCÍCIO

Resolver: $x^3 - x^2 - 8x = 0$

Resp.:

$$0, 0.5 + 0.5\sqrt{33}, 0.5 - 0.5\sqrt{33}$$

Derivação de funções com diff()

```
>> x=sym('x');
>> Deriv=diff('x*sin(x)^2', x)
```

```
Deriv = sin(x)^2+2*x*sin(x)*cos(x)
```

```
pretty(Deriv)
```

```
       2
sin(x)  + 2 x sin(x) cos(x)
```

Derivação de funções com diff()

EXERCÍCIO Calcular a derivada das funções abaixo

x^3-4.5*sin(x)+3.73	**Resp.: 3*x^2-4.5*cos(x)**
x*sin(x)*cos(x)^3	**Resp.: sin(x)*cos(x)^3+x*cos(x)^4-3*x*sin(x)^2*cos(x)^2**

Derivação de funções com diff()

EXERCÍCIO

Definir f(x)=x^3-sin(atan(x))-tan(asin(x)).
Calcular a derivada de menor ordem de f(x) que NÃO seja nula em x=0, e dar esse valor.
Sugerimos traçar o gráfico de f(x) no intervalo -1<= x <=+1.

Resposta:

$$3x^2 - \frac{1}{(1+x^2)^{1/2}} + \frac{x^2}{(1+x^2)^{3/2}} - \frac{1}{(1-x^2)^{1/2}} - \frac{x^2}{(1-x^2)^{3/2}}$$

Derivada de f() em x=0 é -2

Ver pg 16 e 31.

Integração de funções com int()

```
>> x=sym('x');
>> Integ=int('x*sin(x)^2', x)
```

```
Integ = x*(-1/2*cos(x)*sin(x)+1/2*x)+1/4*sin(x)^2-1/4*x^2
```

```
pretty(Integ)
```

```
                                                      2          2
x (- 1/2 cos(x) sin(x) + 1/2 x) + 1/4 sin(x)  - 1/4 x
```

Integração de funções com int()

EXERCÍCIO Calcular as integrais das funções a seguir:

x^3*cos(x) **Resp.:**
x^3*sin(x)+3*x^2*cos(x)-6*cos(x)-6*x*sin(x)

sin(x)^2*x^2 **Resp.:**
???

Integral definida de funções com int()

```
>> x=sym('x');
>> IntegDef=int('1/(1+x^4)',x,1,2)
>> pretty(IntegDef)
```

```
>> NUMERO=numeric(IntegDef)
```

```
NUMERO =    0.2032
```

```
         1/2              1/2          1/2             1/2
    1/8 2    log(5 + 2 2   ) - 1/8 2    log(5 - 2 2   )

                 1/2           1/2               1/2           1/2
         + 1/4 2    atan(2 2    + 1) + 1/4 2    atan(2 2    - 1)

                1/2           1/2          1/2          1/2          1/2
         - 1/8 2    log(2 + 2   ) + 1/8 2    log(2 - 2   ) - 1/8 2    pi
```

EXERCÍCIO Calcular numericamente as integrais a seguir:

$$\int_{-p/2}^{p/2} abs(\sin(x))dx$$

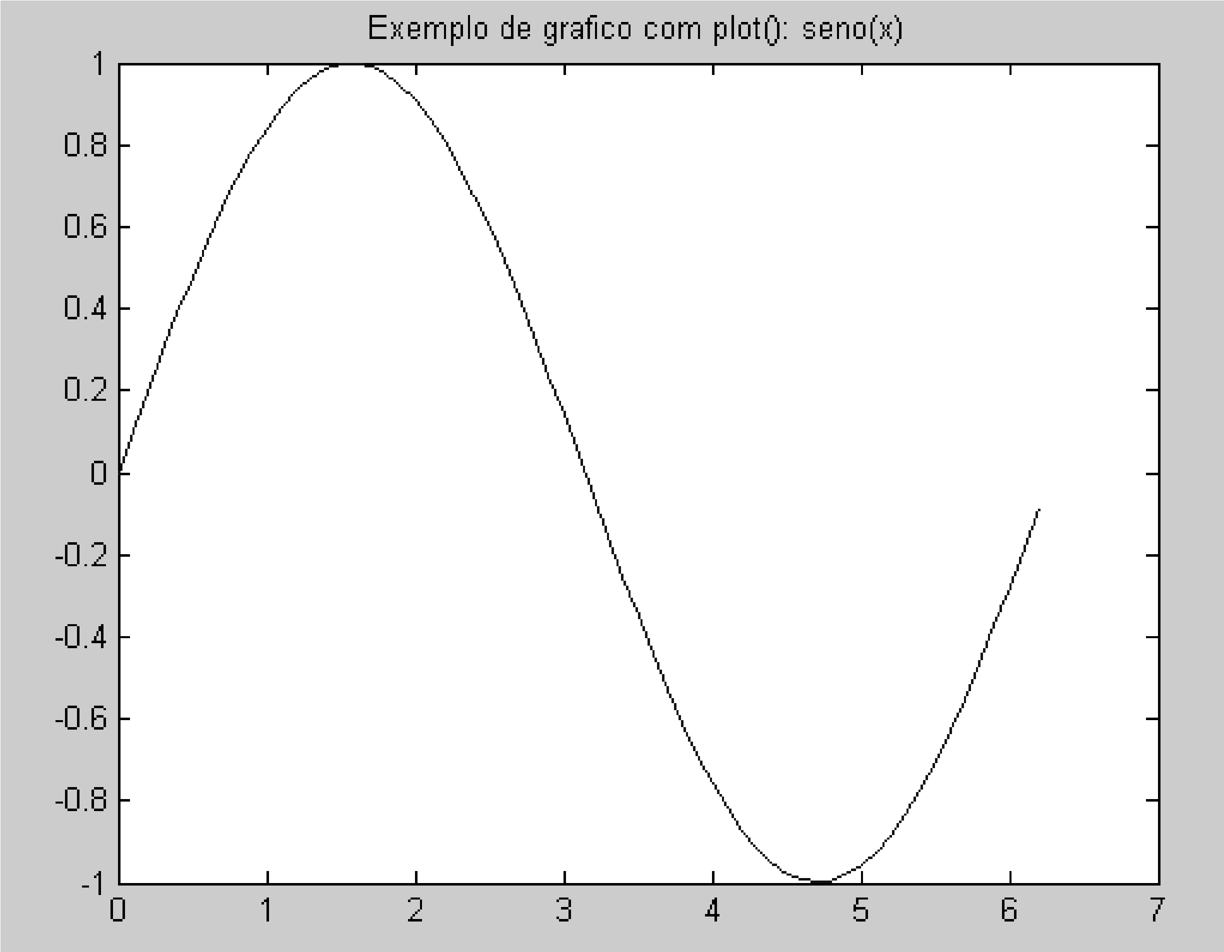
Resp.: 2

$$\int_{0}^{1} \frac{1}{\sqrt{1+x^4}}dx$$

Resp.:0.9270373385

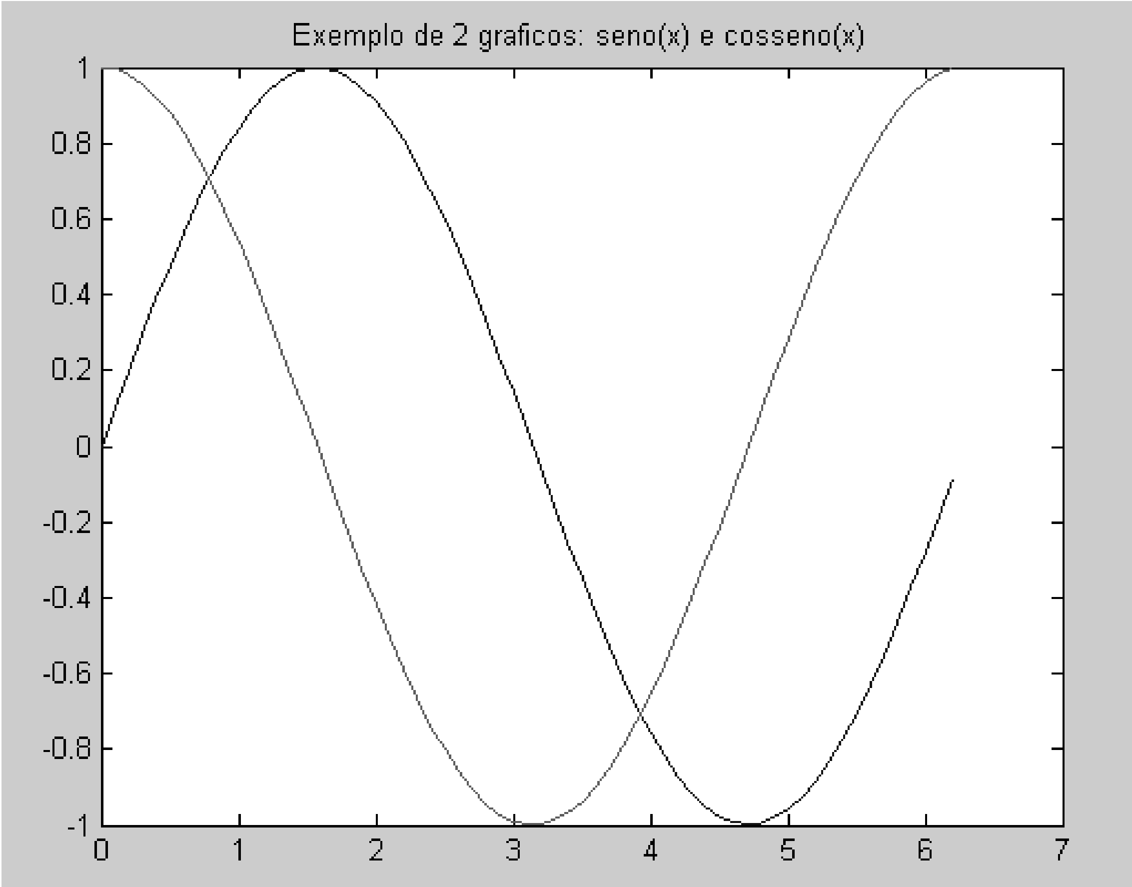
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
plot(x,y)
title('Exemplo de grafico com plot(): seno(x)') % define título
```



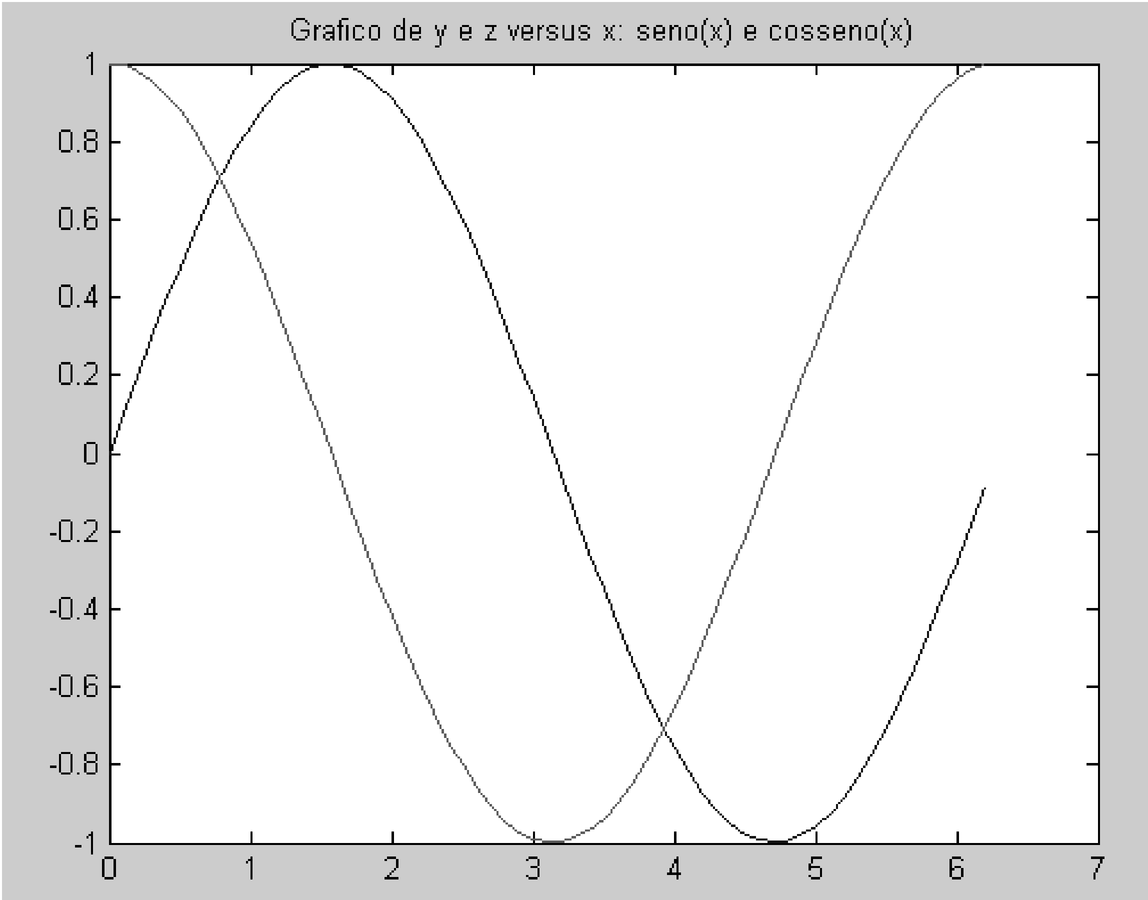
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x) % cosseno de x
plot(x,y,x,z) % dois gráficos
title('Exemplo de 2 graficos: seno(x) e cosseno(x)') % define título
```



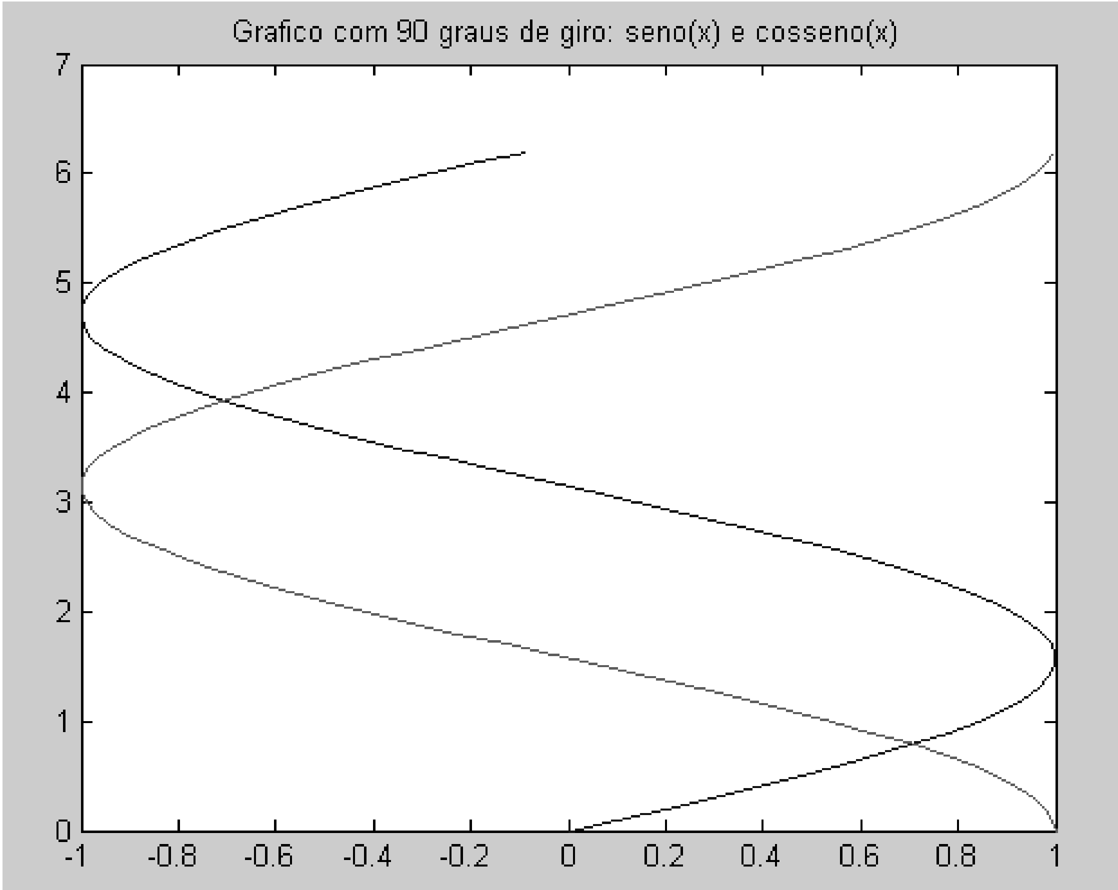
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
Matr=[y;z]; % definir uma matriz com seno e cosseno
plot(x,Matr) %  // gráfico de Matr versus x
title('Grafico de y e z de Matr versus x: seno(x) e cosseno(x)')
                                                        % define título
```



plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
Matr=[y;z] % definir uma matriz com seno e cosseno
plot(Matr,x) %  // matriz como 1o. argumento
title('Grafico com 90 graus de giro: seno(x) e cosseno(x)')
                                             % define título
```



plot()

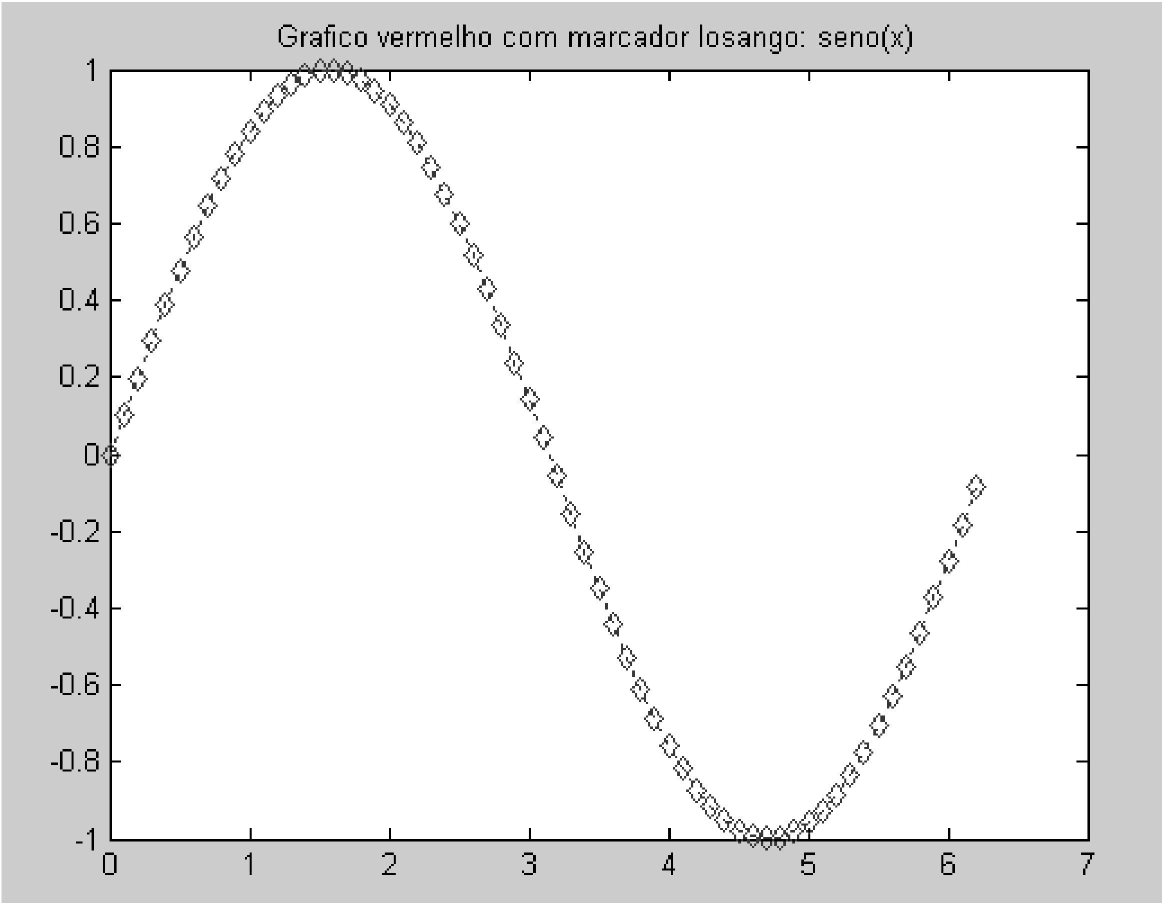
símbolo	cor
b	azul
g	verde
r	vermelho
c	ciano
m	magenta
y	amarelo
k	preto
w	branco

símbolo	marca
.	ponto
o	círculo
x	xis
s	quadrado
d	losango
v	triâng p/ baixo
^	triâng p/ cima
p	pentagrama
h	hexagrama
<	Triâng p/ esq.
>	Triâng p/ dir.

símbolo	Tipo de linha
-	contínua
:	pontilhada
-.	Traço e pto.
--	tracejada

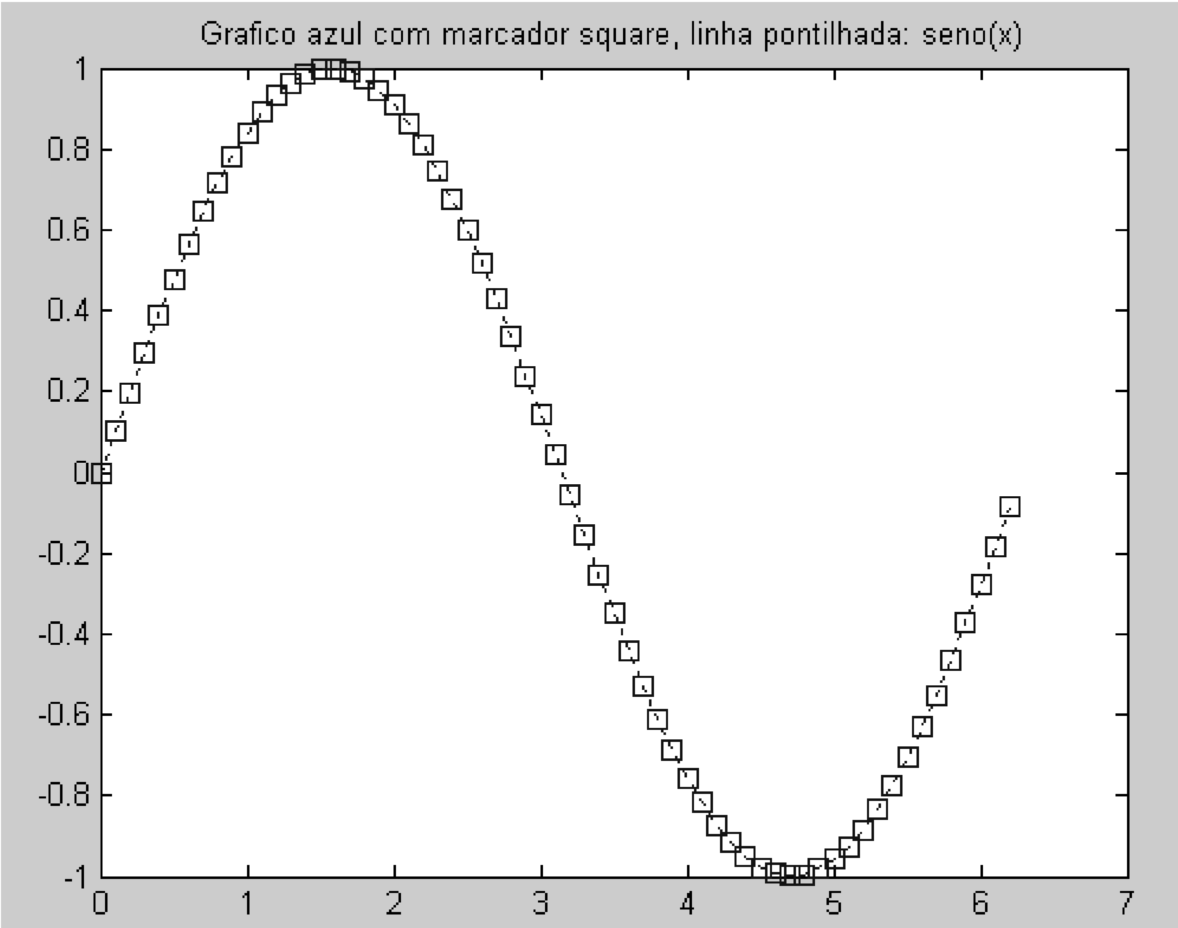
plot()

x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
plot(x,y,'rd') % r de red, e d de losango
title('Grafico vermelho com marcador losango: seno(x)')
% define título



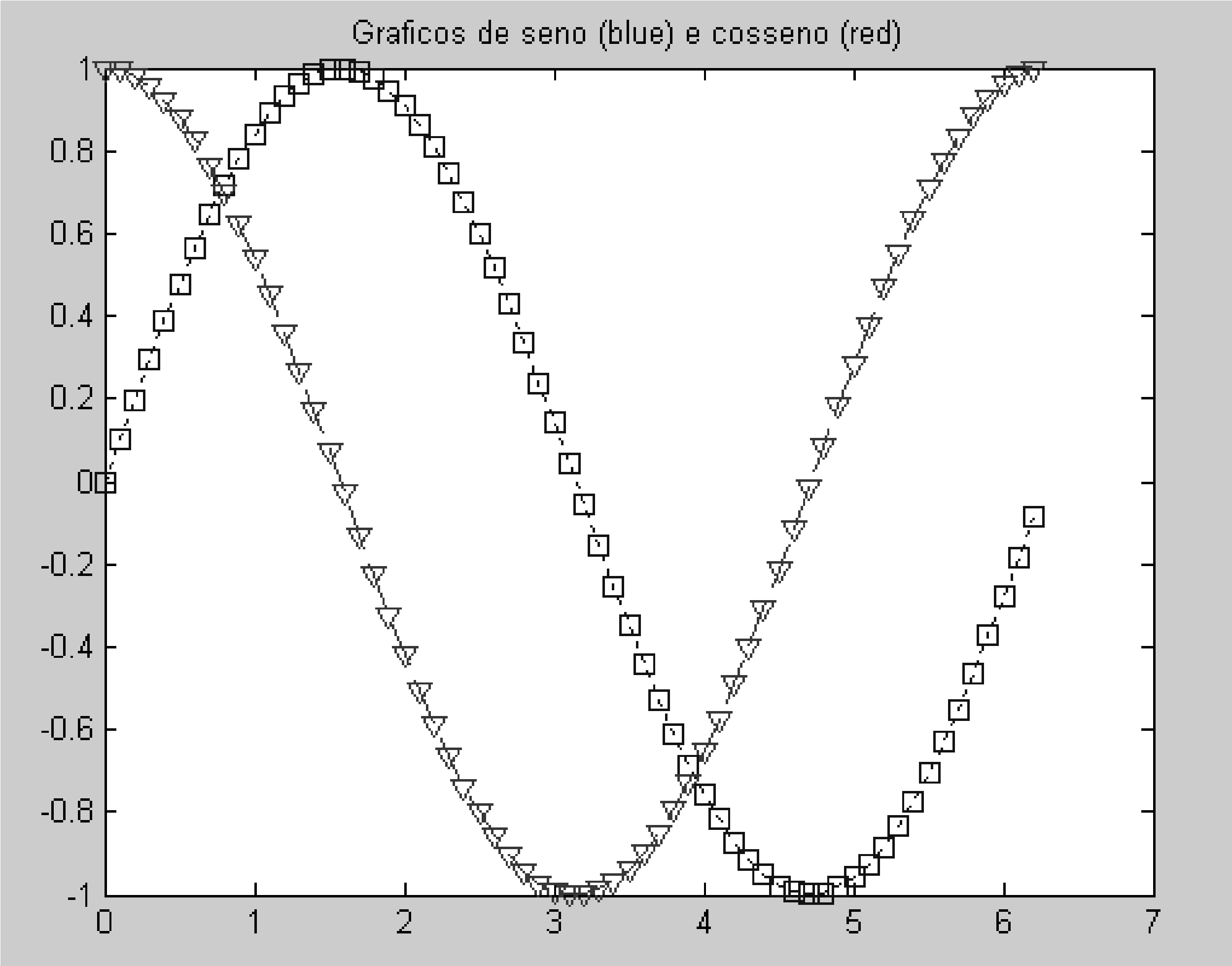
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
plot(x,y,'b:s')
title('Grafico azul com marcador square, linha pontilhada: seno(x)')
```



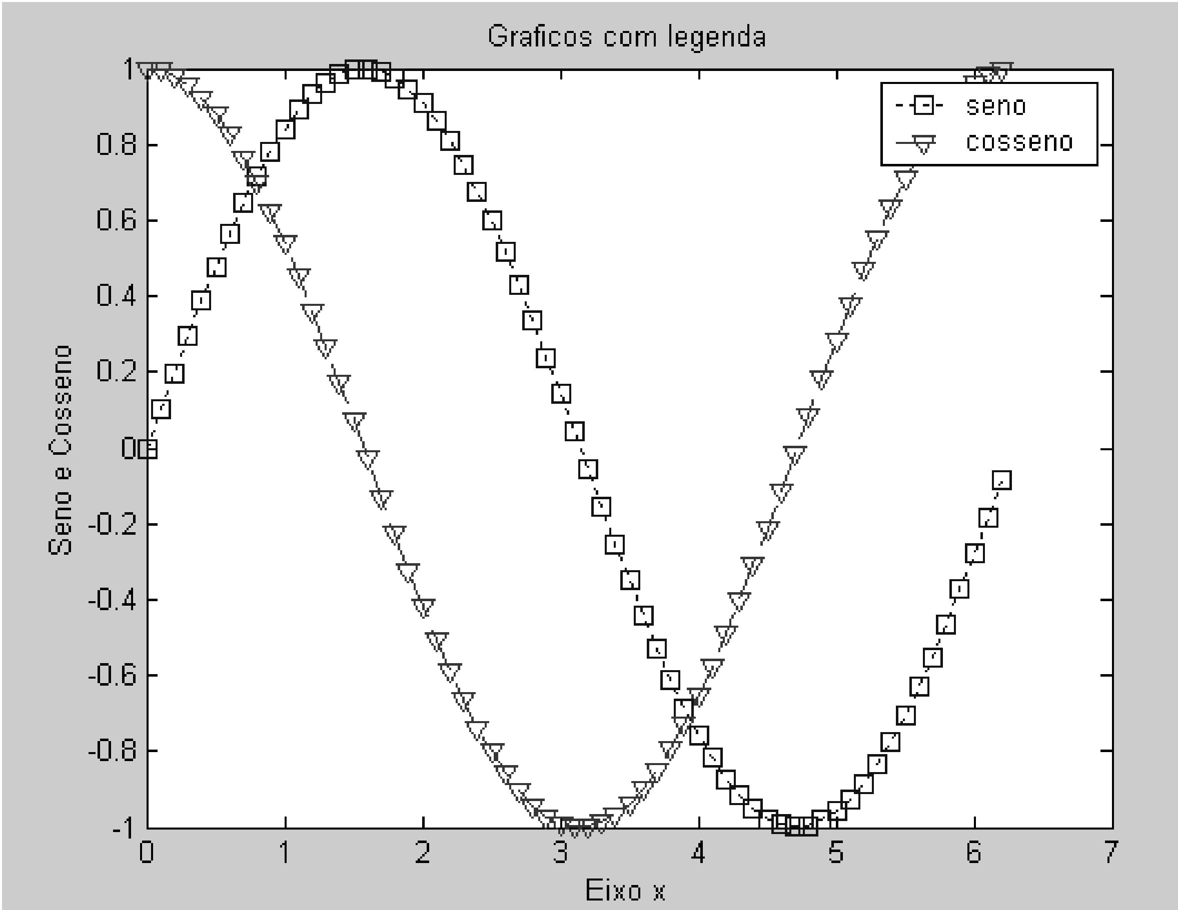
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
plot(x,y,'b:s',x,z,'rv--')
title('Graficos de seno (blue) e cosseno (red)')
```



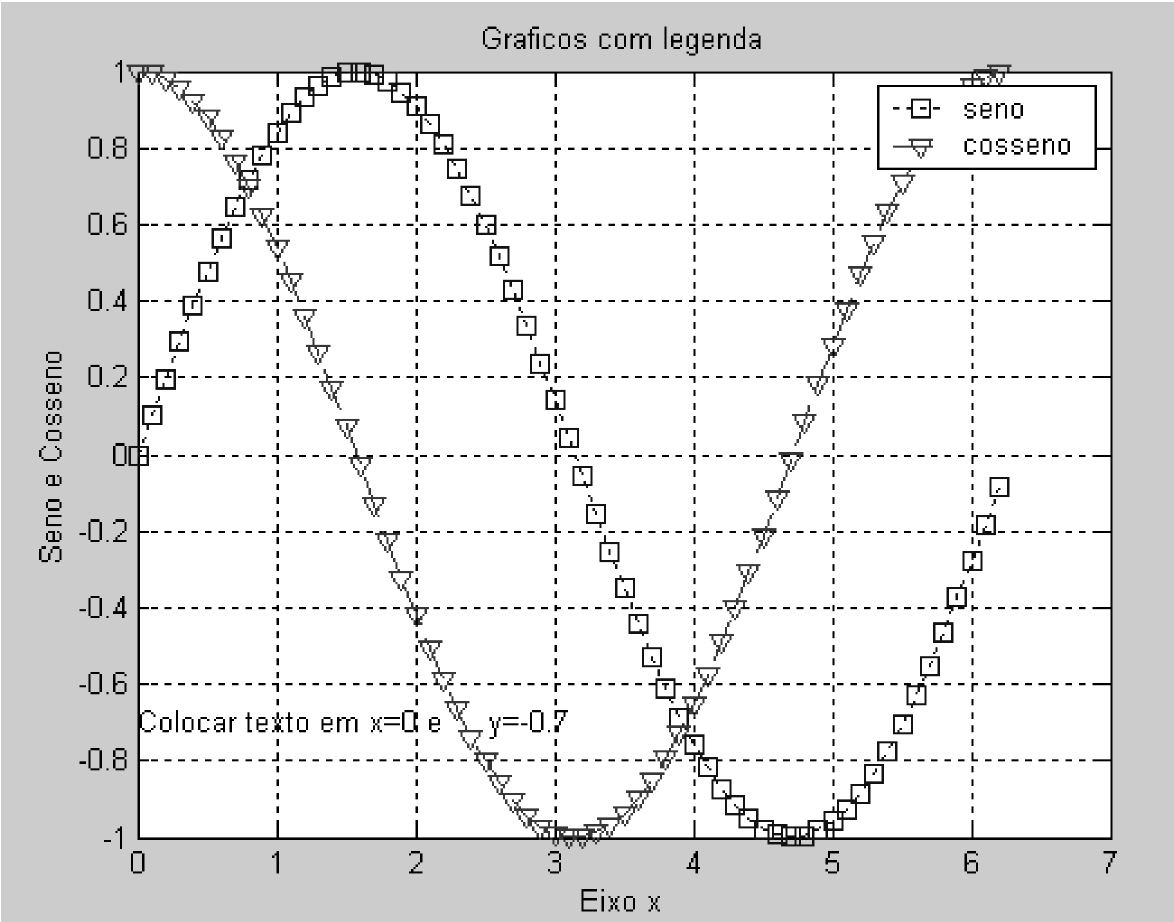
plot()

```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); z=cos(x);
plot(x,y,'b:s',x,z,'rv--')
title('Graficos com legenda')
xlabel('Eixo x') % eixo horizontal
ylabel('Seno e Cosseno') % eixo vertical
legend('seno','cosseno') % inserir legenda, na ordem
%        que pode ser deslocada arrastando-a c/ mouse
```



plot()

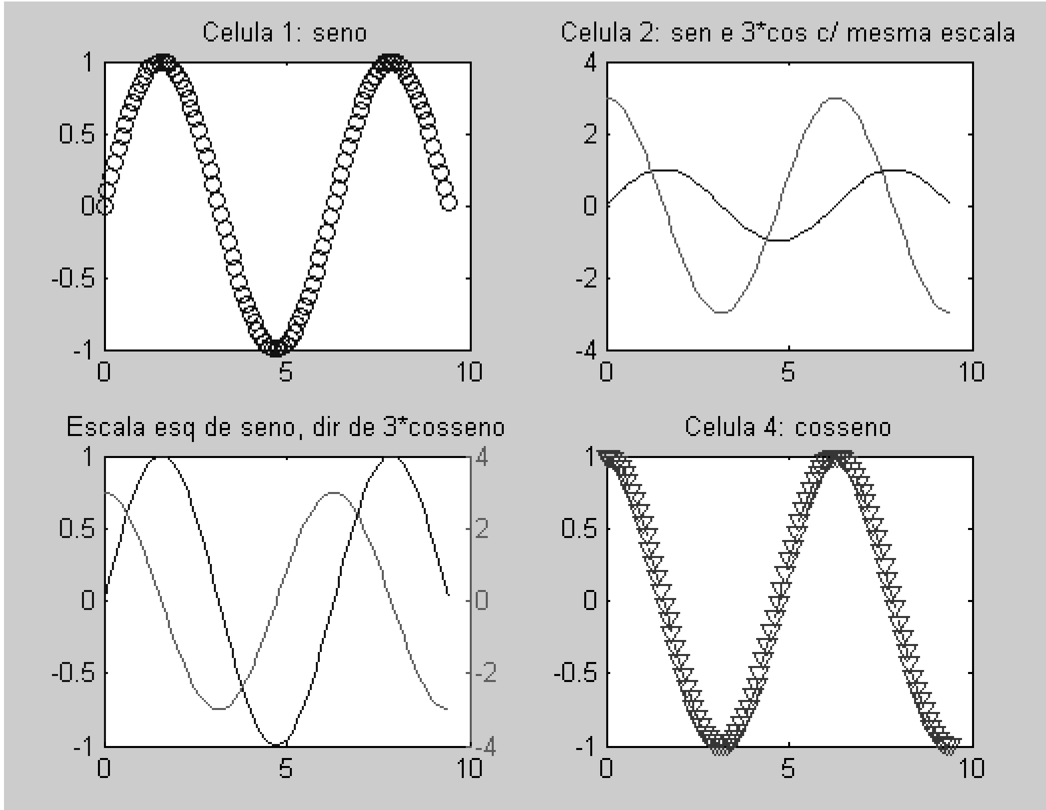
```
x=0:0.1:2*pi; % define pontos no eixo x
y=sin(x); z=cos(x);
plot(x,y,'b:s',x,z,'rv--')
title('Graficos com legenda')
xlabel('Eixo x') % eixo horizontal
ylabel('Seno e Cosseno') % eixo vertical
legend('seno','cosseno') % inserir legenda, na ordem
grid on  % para mostrar reticulado; grid off p/ apagar
text(0,-0.7,'Colocar texto em x=0 e     y=-0.7') % p/ incluir texto
```



subplot()

subplot(m,n,prox) divide a janela de gráficos em m linhas e n colunas, sendo prox a próxima célula a receber o gráfico

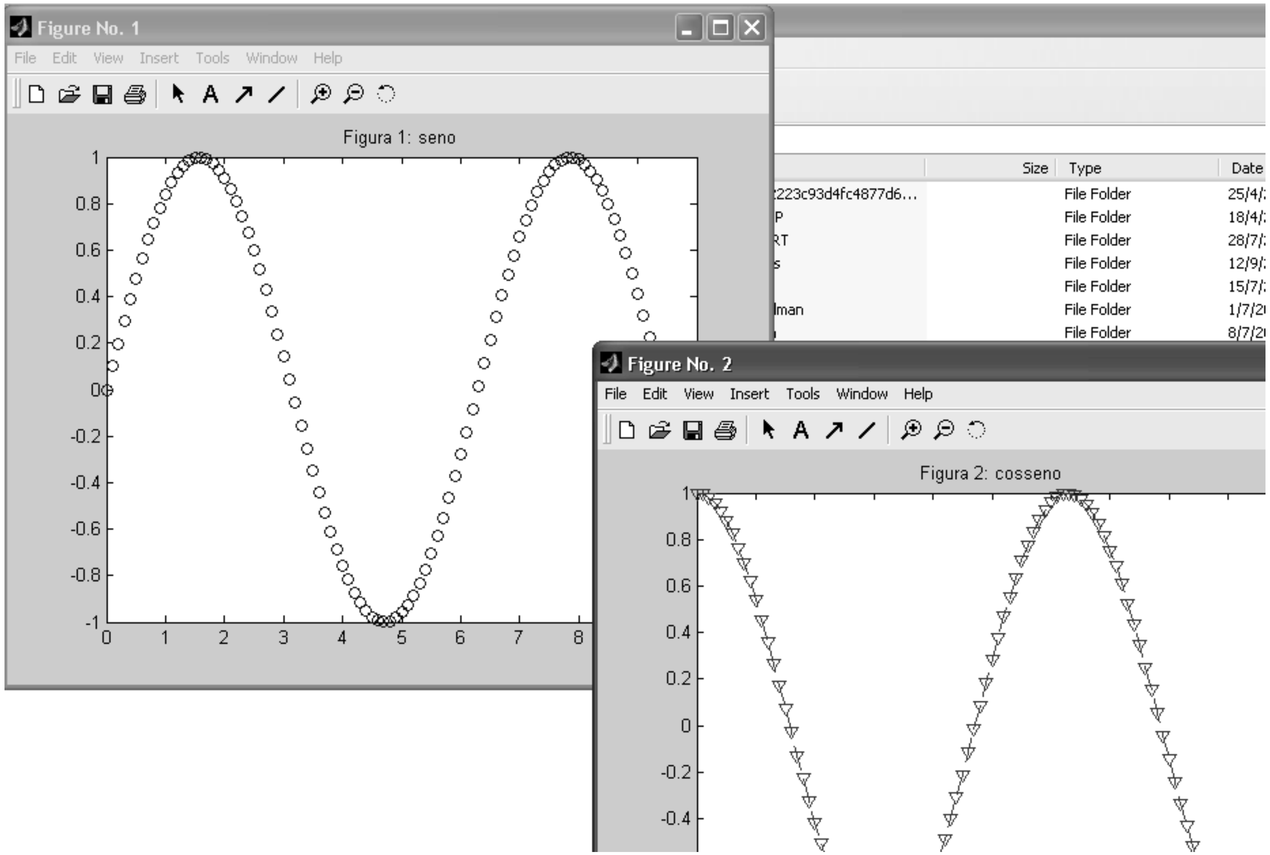
```
x=0:0.1:3*pi; % define pontos no eixo x
y=sin(x); % seno de x
z=cos(x); % cosseno de x
w=3*cos(x);
%%%%%%%%%%%
subplot(2,2,1)
plot(x,y,'bo')
title('Celula 1: seno')
%%%%%%%%%%%
subplot(2,2,4)
plot(x,z,'rv--')
title('Celula 4: cosseno')
%%%%%%%%%%%
subplot(2,2,2)
plot(x,y,x,w)
title('Celula 2: sen e 3*cos c/ mesma escala')
%%%%%%%%%%
subplot(2,2,3)
% plotyy p/ ter escala distinta nos eixos verticais
% escala de y no eixo vert esquerdo,
% de w no vert direito
plotyy(x,y,x,w)
title('Escala esq de seno, dir de 3*cosseno')
```



plotyy()

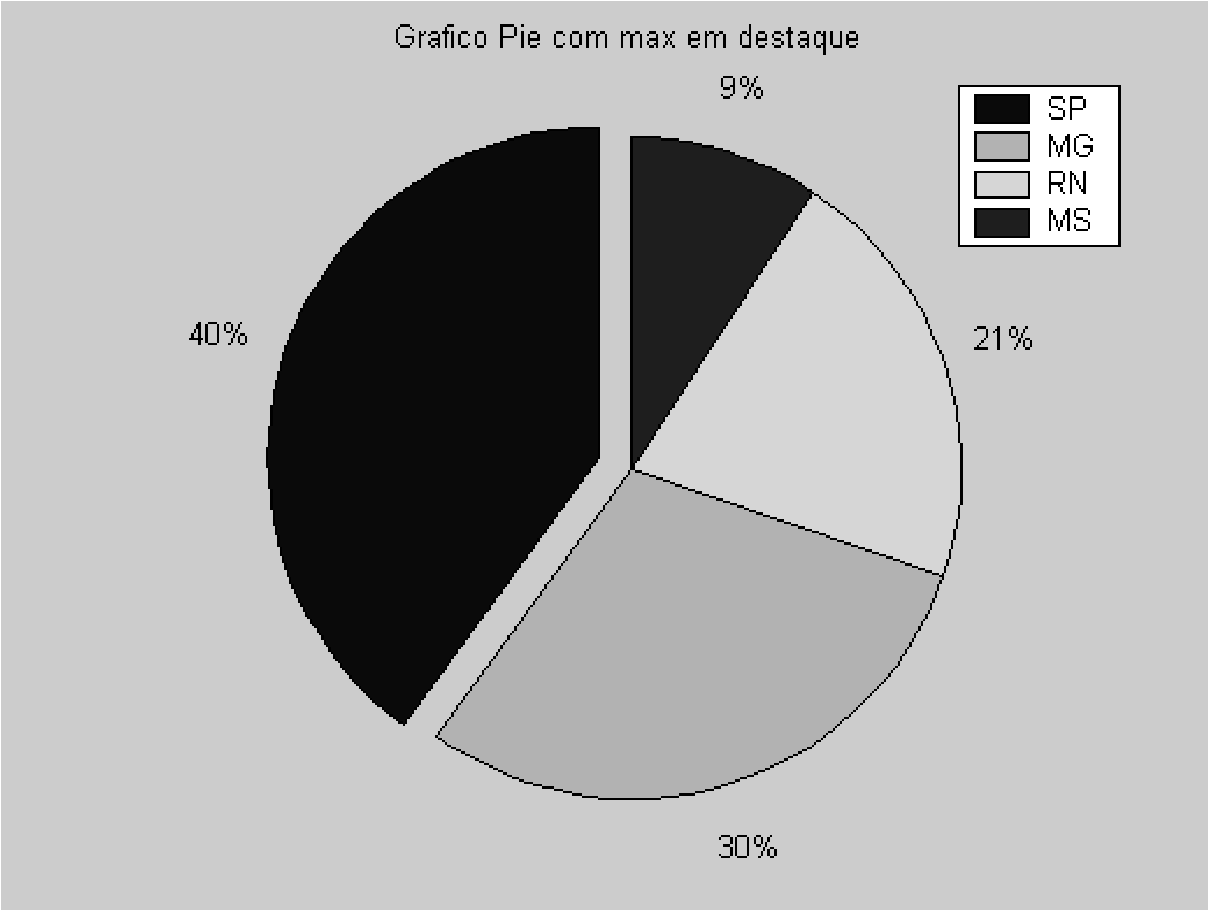
```
y=sin(x); % seno de x
z=cos(x); % cosseno de x
%%%%%%%%%%%%%%%%%%%%
figure(1) % próx gráfico na janela 1
plot(x,y,'bo')
title('Figura 1: seno')
%%%%%%%%%%%%%%%%%%%%
figure(2) % próx gráfico na janela 2
plot(x,z,'rv--')
title('Figura 2: cosseno')
```

figure()



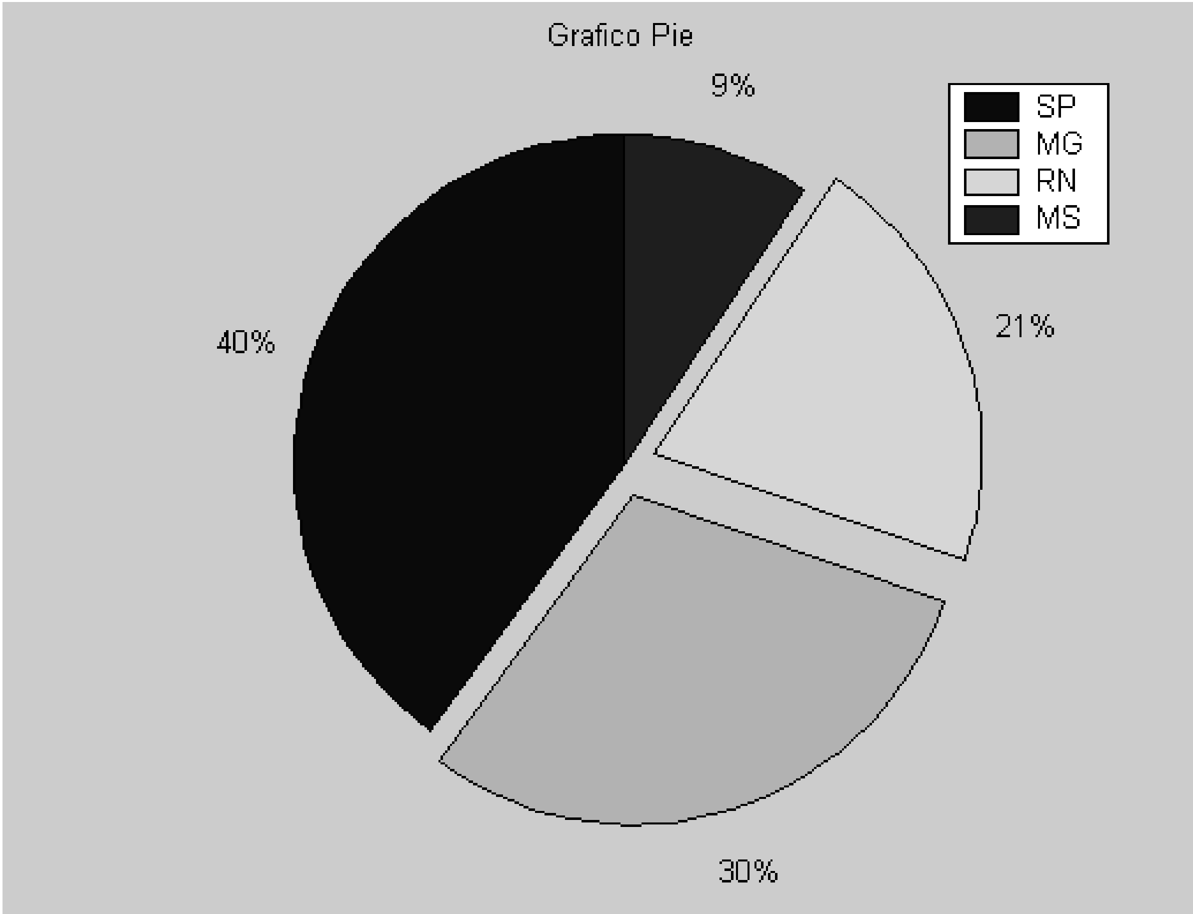
pie()

```
A=[4.3 3.2 2.25 1];
pie(A,A==max(A)); % destaca a fatia maior
title('Grafico Pie com max em destaque')
legend('SP','MG','RN','MS')
```



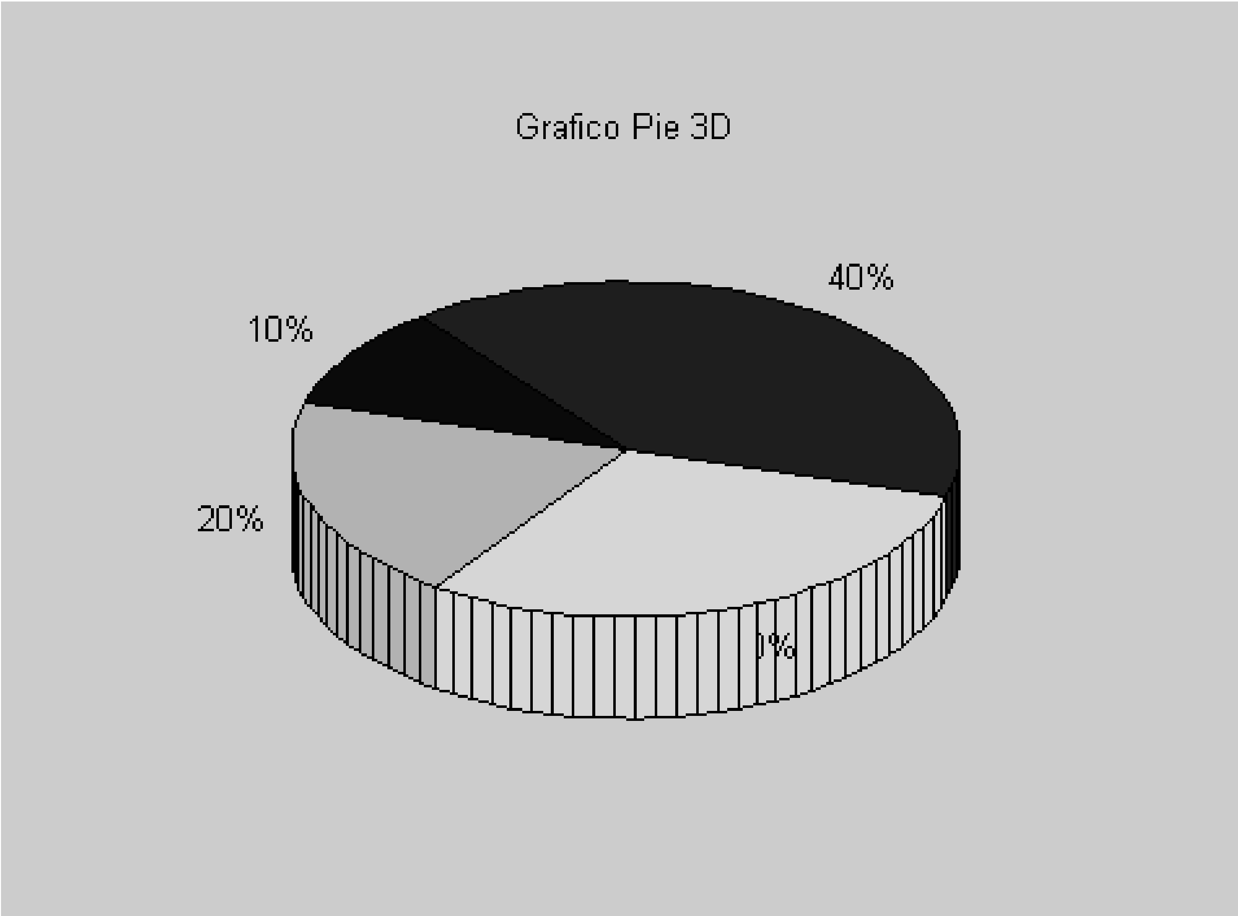
pie()

```
A=    [4.3 3.2 2.25 1];
pie(A,[0    1   1     0]) % destaca as fatias com 1 na posicao correspondente
title('Grafico Pie')
legend('SP','MG','RN','MS')
```



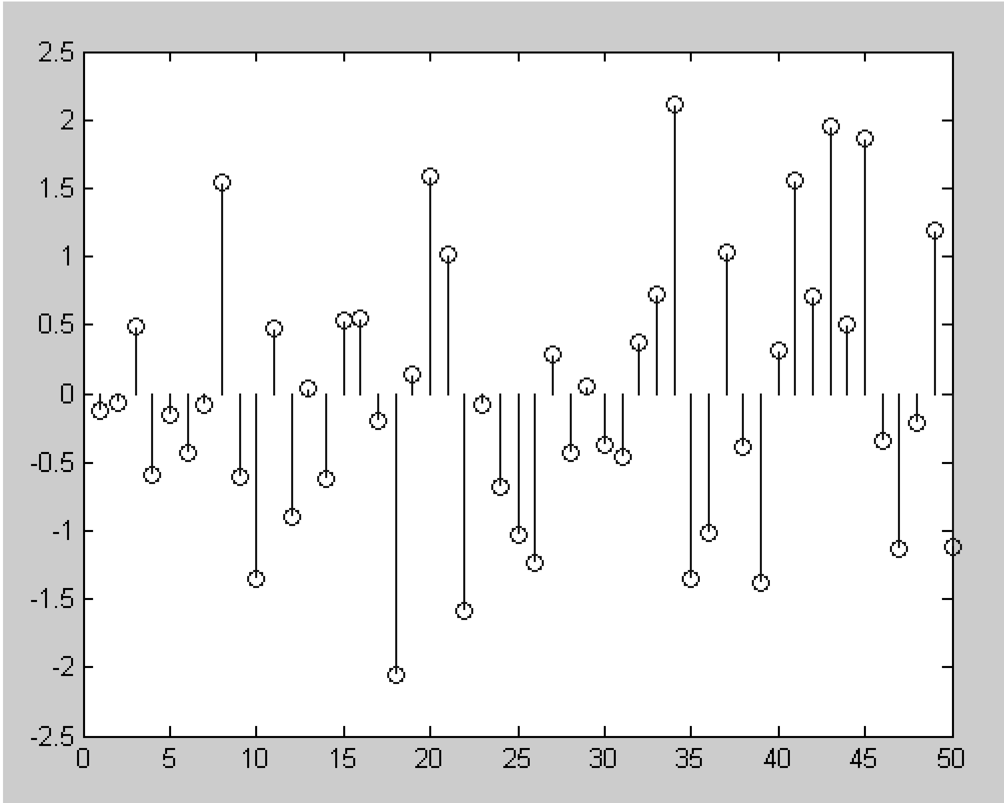
pie3()
p/ 3D

A= [1.1 2.2 3.3 4.4];
pie3(A)
title('Grafico Pie 3D')



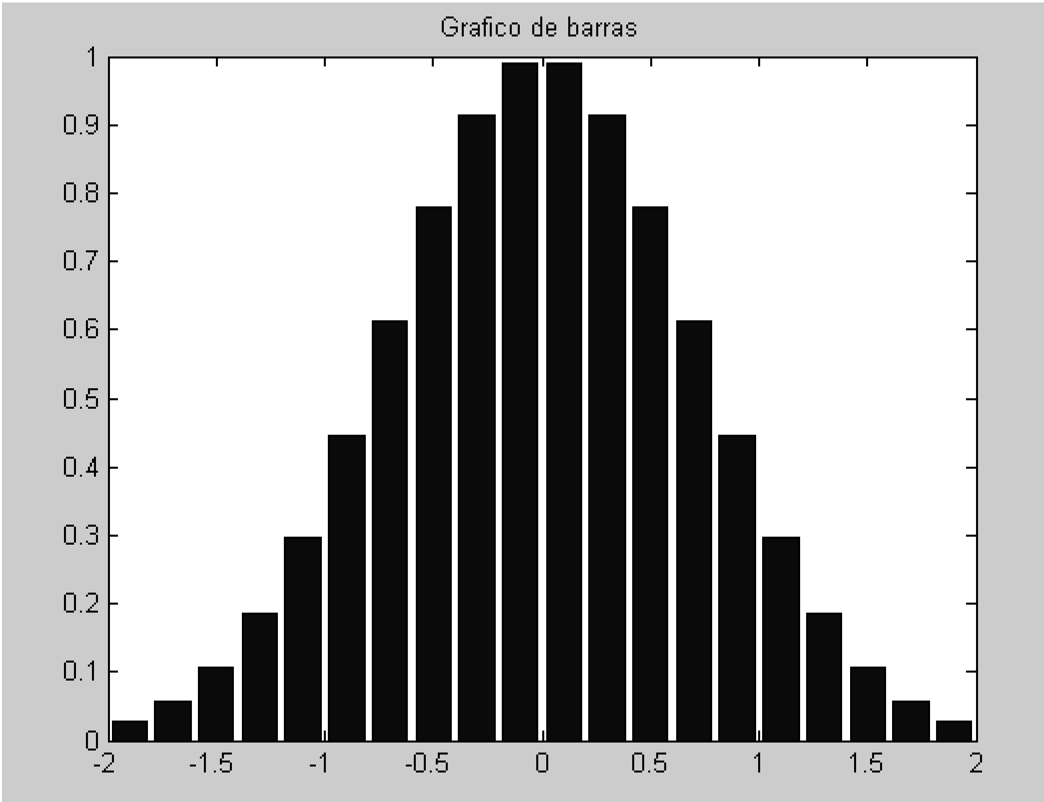
stem()

% gera 50 valores, 1 coluna, distribuicao normal
% media zero, variancia 1
norma=randn(50,1)
stem(norma,'o') % mostra 50 hastes



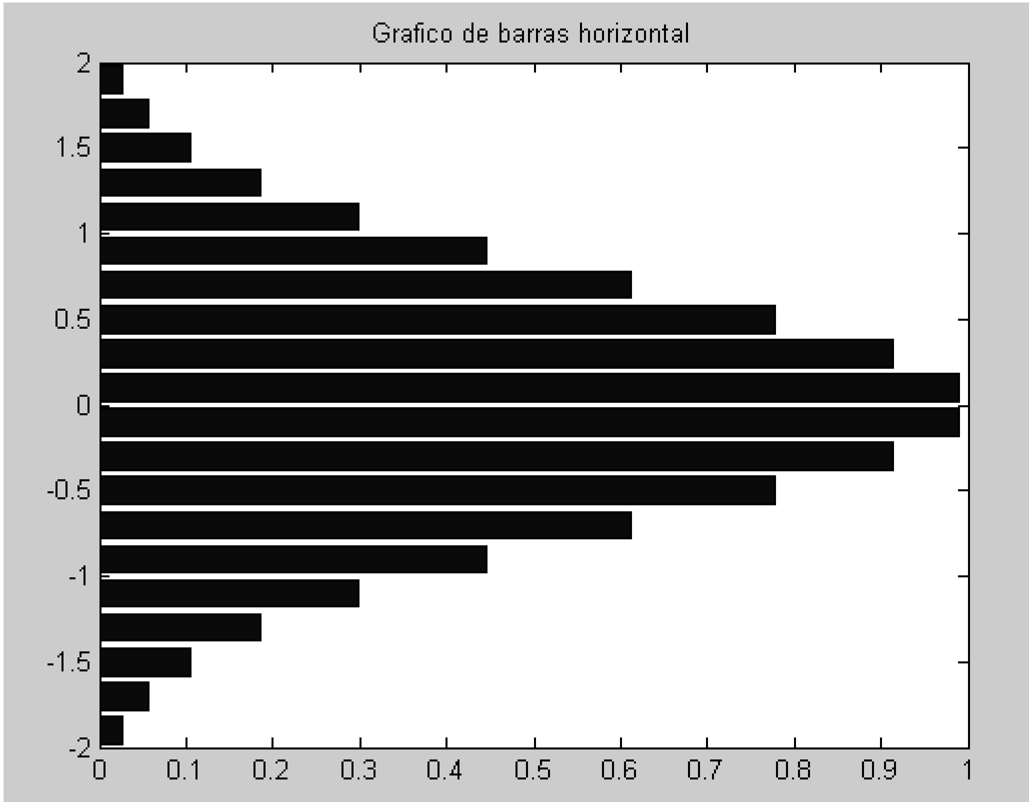
```
x=-1.9:0.2:1.9; % cria x
y=exp(-x.*x); % cria y
bar(x,y)
title('Grafico de barras')
```

bar()



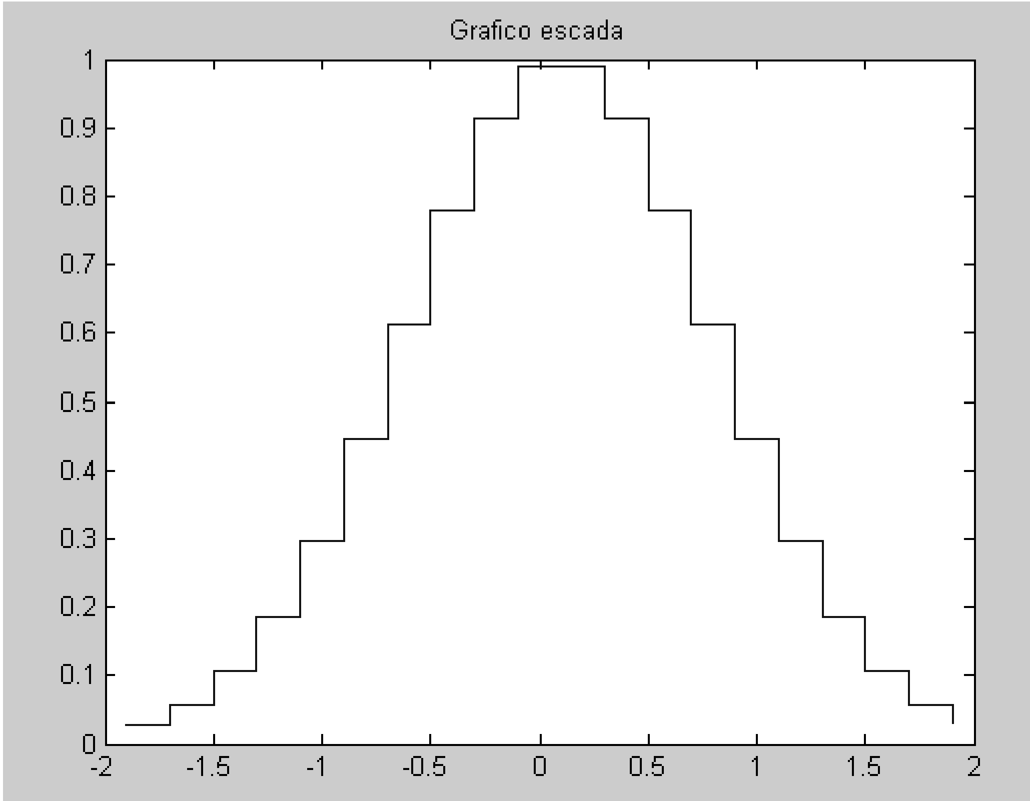
barh()

```
x=-1.9:0.2:1.9; % cria x
y=exp(-x.*x); % cria y
barh(x,y)
title('Grafico de barras horizontal')
```



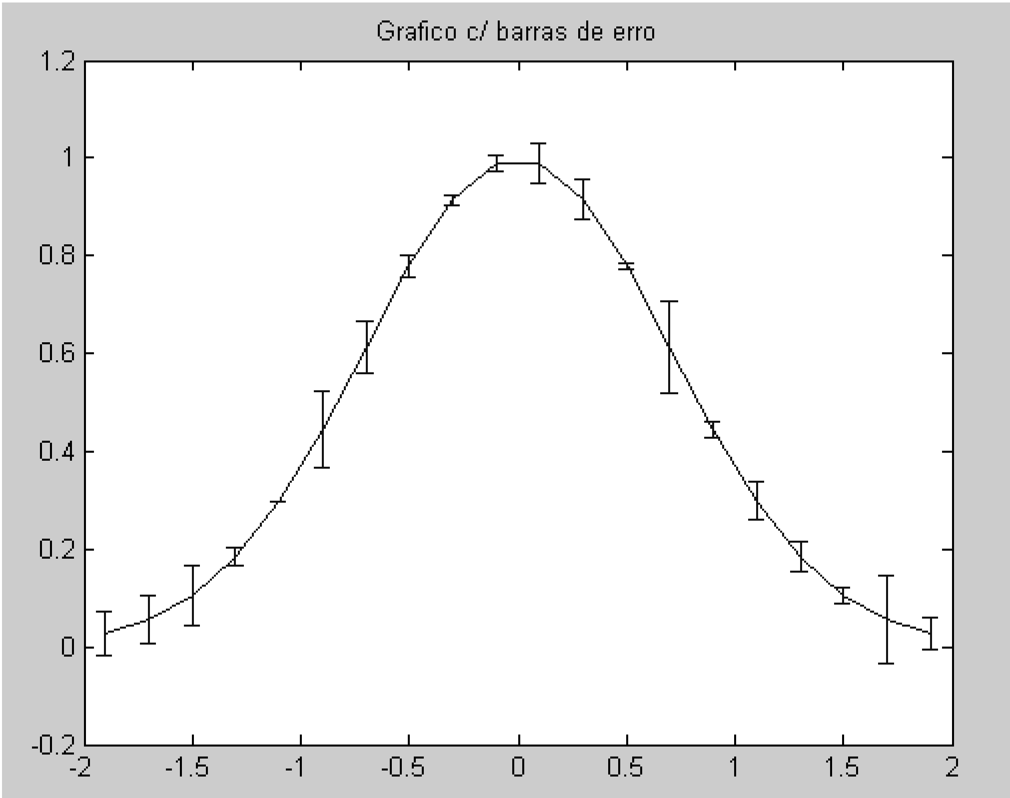
stairs()

x=-1.9:0.2:1.9; % cria x
y=exp(-x.*x); % cria y
stairs(x,y)
title('Grafico escada')



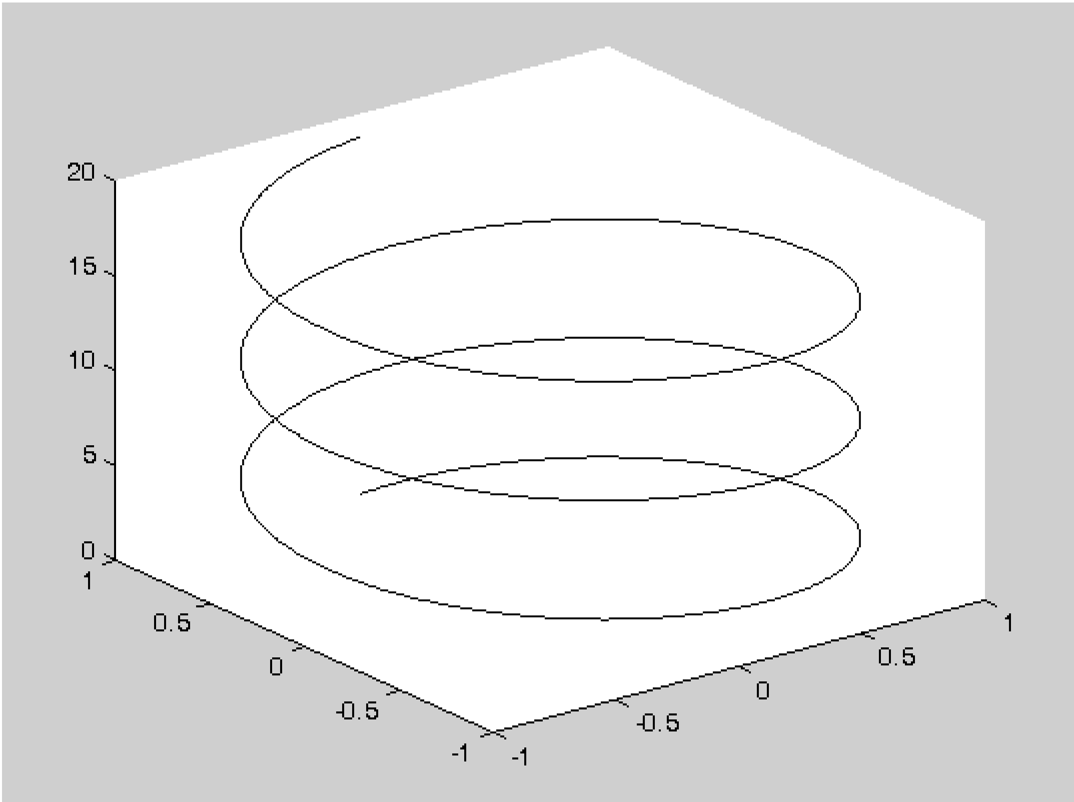
```
x=-1.9:0.2:1.9; % cria x
y=exp(-x.*x); % cria y
e=rand(size(x))/10 % pseudo aleatório
errorbar(x,y,e) % barra com y+e, y-e
title('Grafico c/ barras de erro')
```

errorbar()



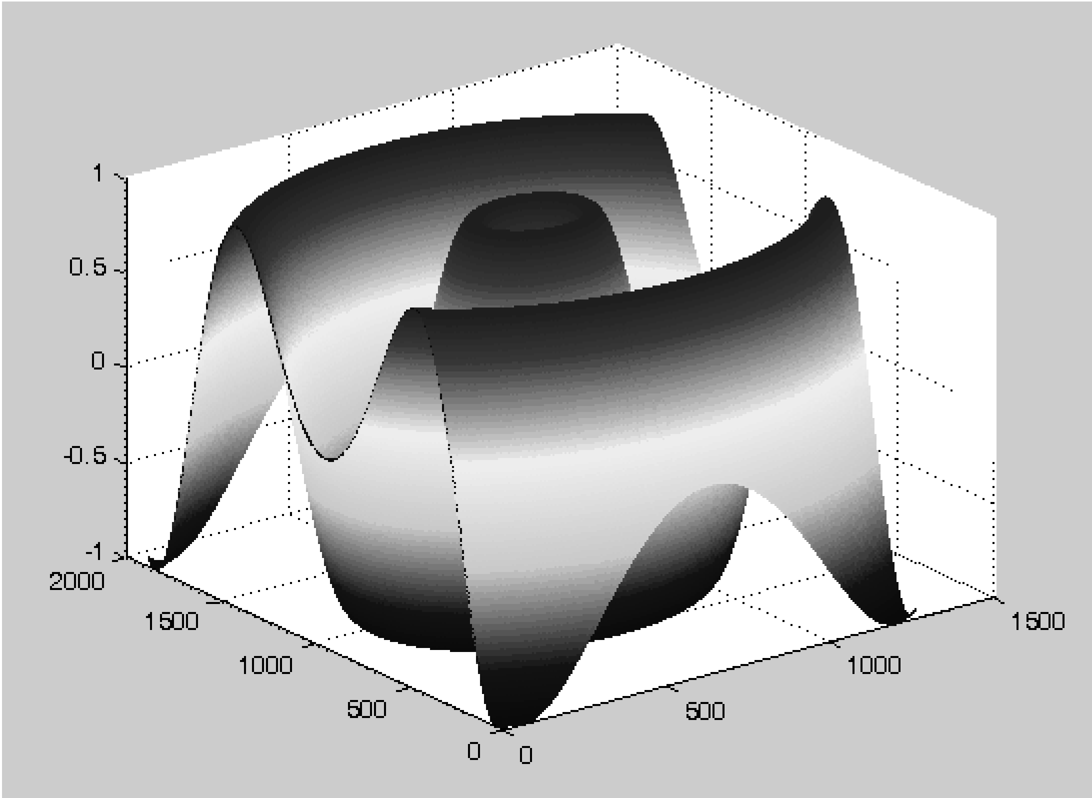
plot3()

```
>> % plot3, helice (sen(t),cos(t),t)
>> gradet=0:0.01:6*pi; % intervalo para eixo t
>> plot3(sin(gradet), cos(gradet), gradet)
```



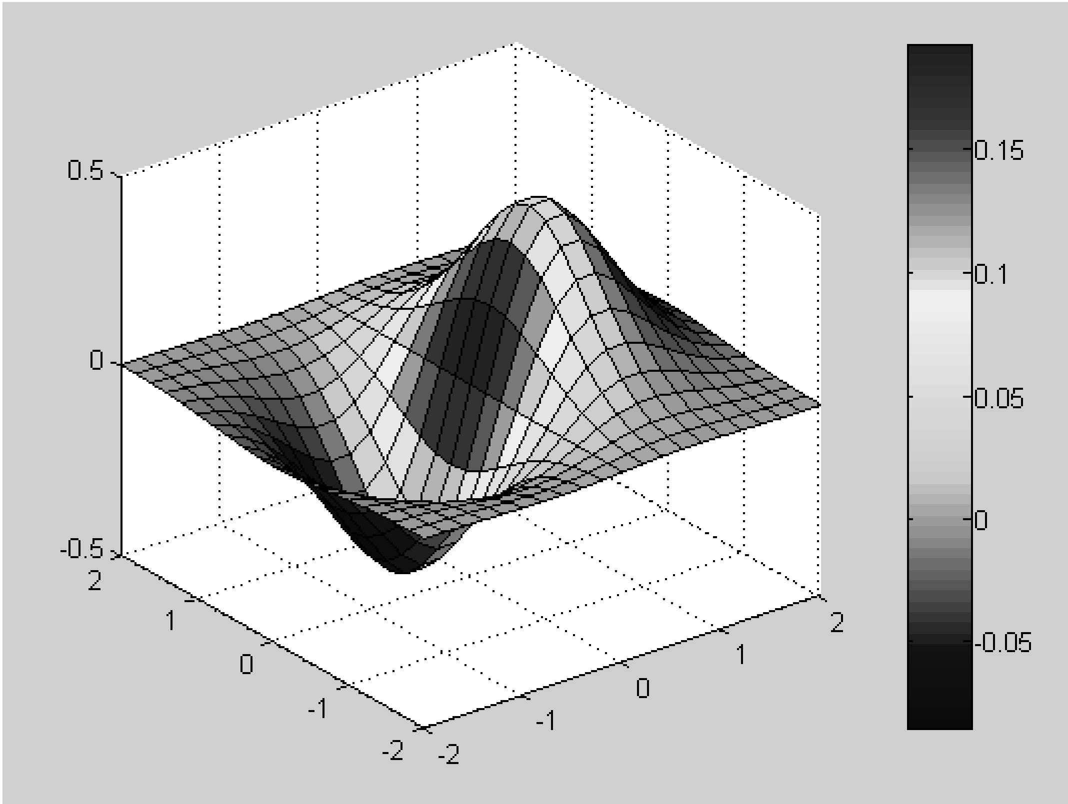
mesh()

```
>> % definir um dominio X Y
>>[X, Y]= meshgrid(-2*pi:0.01:2*pi, -3*pi:0.01:3*pi); % note o ;
>> mesh( sin(sqrt(X.*X+Y.*Y)) )
```



surf()

```
>>[x,y]= meshgrid([-2:.2:2]);
>> Z= x.*exp(-x.^2-y.^2);
>> surf(x,y,Z,gradient(Z))
>> colorbar
```



Solução de equações diferenciais com dsolve()

$$x''(t) + 2x(t) = 0, x(0) = 0, x'(0) = 1$$

>> Sol=dsolve('D2x+2*x=0', 'x(0)=0, Dx(0)=1')

Sol = 1/2*sin(t*2^(1/2))*2^(1/2)

```
                1/2   1/2
1/2 sin(t 2   ) 2
```

Solução de equações diferenciais com dsolve()

$$x''(t) + 2x(t) = 0, x(0) = 0, x(3) = 1$$

```
>> Sol2=dsolve('D2x+2*x=0', 'x(0)=0, x(3)=1')
```

```
Sol2 =  -1/sin(2^(1/2))/(-3+4*sin(2^(1/2))^2)*sin(t*2^(1/2))
```

```
                   1/2
              sin(t 2   )
    - ----------------------------
             1/2                1/2 2
        sin(2   ) (-3 + 4 sin(2   ) )
```

EXERCÍCIO Resolver as equações diferenciais a seguir:

$$\frac{dy}{dt} + 2y = 0 \qquad\qquad y(t) = 3e^{-2t}$$

$$m\frac{d^2y}{dt^2} = mg,\, y(0) = h, \frac{dy}{dt}(0) = 0 \qquad y(t) = -gt^2 + h$$

h **é altura de lançamento de um corpo de massa *m*,**
g **é constante de gravidade**

Equação de Cauchy-Euler

$$t^2x'' - 2tx' + 3x = 0 \text{ (Cauchy-Euler)}$$

```
>> CEuler=dsolve('t^2*D2x-2*t*Dx+3*x=0')
>> pretty(CEuler)
```

```
CEuler =
C1*t^(3/2)*cos(1/2*3^(1/2)*log(t))+C2*t^(3/2)*sin(1/2*3^(1/2)*log(t))



          3/2           1/2                    3/2           1/2
    C1 t    cos(1/2 3    log(t)) + C2 t    sin(1/2 3    log(t))
```

Equação não-linear (Runge-Kutta)

A função ode45 () permite resolver equações diferenciais pelo Método de Runge-Kutta. Exemplificamos a seguir com a resolução da seguinte equação não-linear de pêndulo forçado $x(t)$:

$$x'' + 0.1x' + sen(x) = 0.02\cos(t), x(0) = 0, x'(0) = 1$$

Primeiro convertemos esta equação para um sistema de equações de primeira ordem:

$$\begin{cases} x' = y \\ y' = -0.1y - sen(x) + 0.02\cos(t) \\ x(0) = 0, y(0) = 1 \end{cases}$$

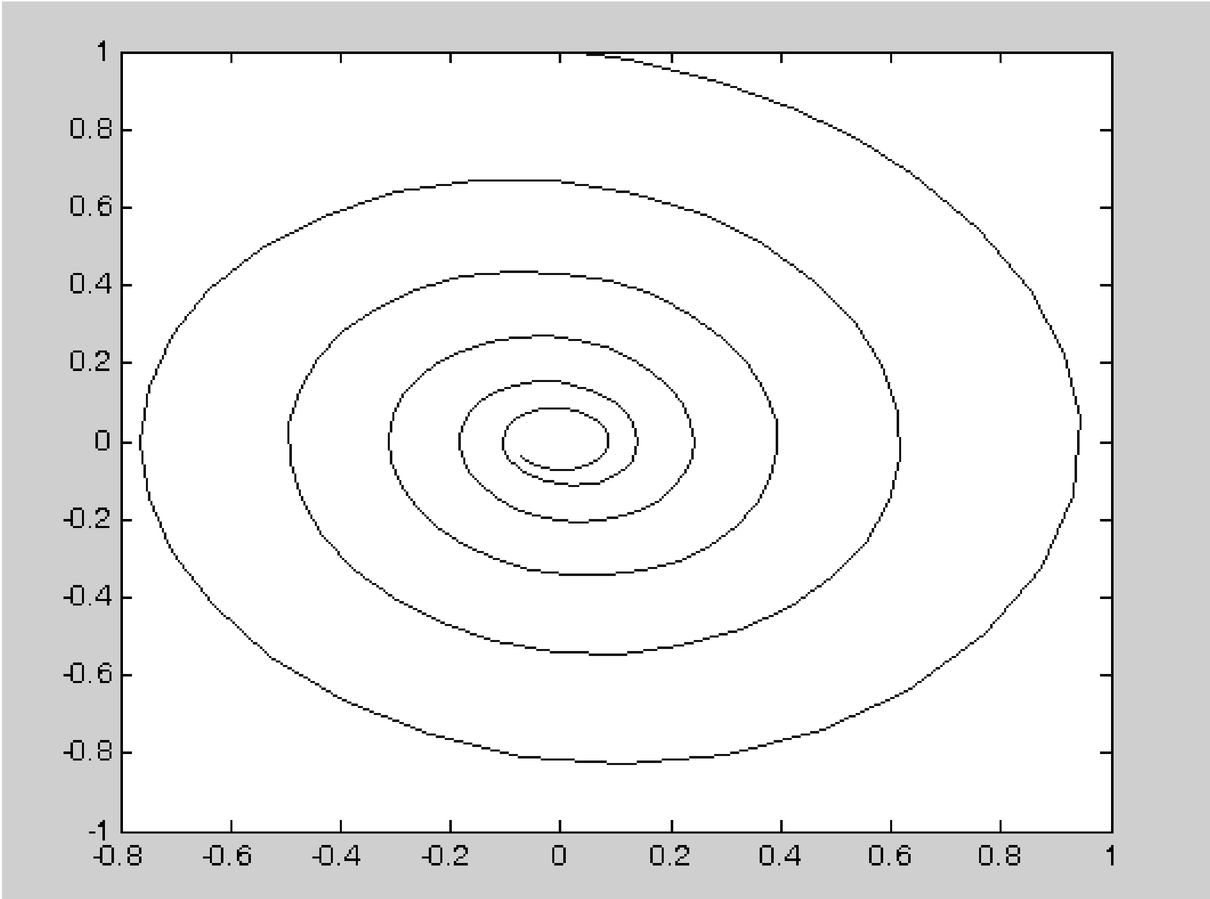
```
function [zaux]=pendulo(taux,z)
% instante taux (valor escalar
% vetor linha z tal que
%       z(1) representa x, e z(2) representa y=x'
% zaux calculado abaixo e' vetor coluna
zaux=[z(2); -0.1*z(2)-sin(z(1))-0.02*cos(taux)];
%      [ y  ;         y           x ]
```

Primeiro, definir arquivo pendulo.m, a ser usada a seguir.

```
>> [t w]=ode45('pendulo',[0 12*pi],[0 1]) % [0 12*pi] e' tempo,
>> plot(w(:,1),w(:,2))                     % [0 1] e' x(0) e y(0)
```

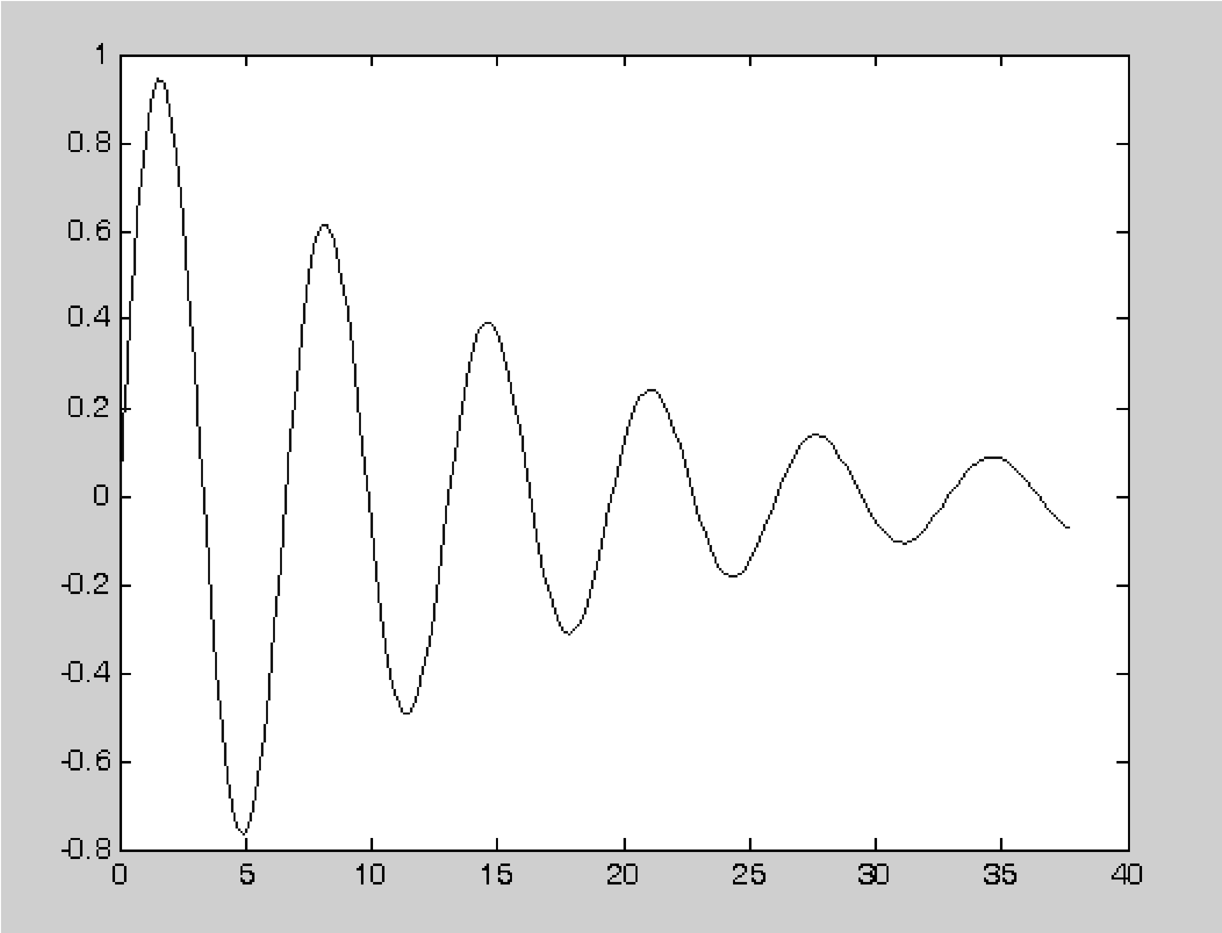
coluna w(:,1) contém valores de x, coluna w(:,2) contém valores de x'

Runge-Kutta



>> plot(t,w(:,1)) % gráfico de x versus t

Runge-Kutta



Transformada de Laplace

Sendo $f(t)$ uma função contínua por partes no intervalo $[0, \infty]$, a transformada de Laplace é definida como a integral abaixo, se a integral existir:

$$L[f](s) \equiv \int_0^{\infty} e^{-st} f(t)dt = \lim_{M \to \infty} \int_0^{M} e^{-st} f(t)dt$$

```
>> syms t s;
>> f='exp(a*t)*exp(b*t)'; % define a funcao f(t)
>> lapla= laplace(f, t, s)
```

```
lapla = 1/(s-a-b)
```

Transformada Inversa

```
>> lapla2= laplace('3*t+1', t, s)
>> invLapla= ilaplace(3/s^2+1/s)
```

```
lapla2 = 3/s^2+1/s
invLapla = 3*t+1
```

Convolução

Sejam $f()$ e $g()$ duas funções sobre um domínio comum $t > 0$. A convolução $f \boxtimes g$ de $f()$ e $g()$ é definida como

$$h(t) = (f \boxtimes g)(t) = \int_0^t f(t-\tau)g(\tau)d\tau.$$

```
>> syms t s
>> Lt= laplace(t, t, s) % transformada de Laplace de t
>> Lt2= laplace(t^2, t, s) % transformada de Laplace de t^2
>> Lprod= Lt*Lt2  % produto das transformadas
>> InvLapla= ilaplace(Lprod)
```

```
Lt =1/s^2
Lt2 =2/s^3
Lprod =2/s^5
InvLapla =1/12*t^4
```

Propriedade importante

$$L[f \boxtimes g] = L[f]\,L[g]$$

$$f \boxtimes g = L^{-1}[L[f]\,L[g]]$$