Primeiros passos no MatLab

Prof. Doherty Andrade - www.metodosnumericos.com.br

1. **Introdução**

O MatLab (Matrix Laboratory) é um software de Matemática e uma linguagem de programação que fornece um bom ambiente de trabalho para todos os usuários. Ele tem três janeals principais:

(a) janela de comandos (command window)- utilizada para inserir comandos e dados.

(b) janela de gráficos - utilizada para exibir gráficos

(c) Janela de edição- utilizada para editar scripts, funtions e **Live Script**.

Neste nestas vamos aprender principalmente a utilizar **Live Script,** uma opção mais ou menos nova do MatLab que fornece várias vantagens ao usuário, pois apresenta os resultados imediatamente abaixo dos comandos ou ao lado se preferir.

2. Comandos básicos

```
35-6
```

```
ans = 29
```

```
sqrt(2)*pi
```

```
ans = 4.4429
```

```
8/2 %cuidado com a barra invertida
```

```
ans = 4
```

```
ans  % ultima resposta
```

```
ans = 4
```

```
A=10,B=30,
```

```
A = 10
B = 30
```

```
A+B,A/B, A*B
```

```
ans = 40
ans = 0.3333
ans = 300
```

```
A=[ 1 2 3; 4 5 6; 7 8 9] % matriz
```

```
A =
     1     2     3
     4     5     6
     7     8     9
```

•

```
b=[ 2 4 6 8]' % vetor coluna - observe a transposta
```

```
b =
     2
     4
     6
     8
```

•

```
who % variaveis utilizadas
```

```
Your variables are:

A    B    ans  b
```

```
!explorer  %abre o explorer do windows
```

```
x=1:10  %operador dois pontos
```

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

•

```
y=1:0.5:4
```

```
y =
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000
```

```
y(2:4) % extraindo alguns elementos de y
```

```
ans =
    1.5000    2.0000    2.5000
```

```
linspace(0,1,5) % linspace(x1,x2,n) gera n pontos igualmente espaçados entre x1 e x2, se n for
```

```
ans =
         0    0.2500    0.5000    0.7500    1.0000
```

```
F='Jota'; %strings
G='Ana';
```

Operações Matemáticas

^ para incluir potência

+ para adicao

- para subtração

/ para divisão

Como definir uma função no MatLab

O MatLab já tem muitas funções definidas. Por exemplo, $\sqrt{x}, \sin(x), \cos(x), \exp(x), \ln(x)$.

Mas podemos definir outras que precisarmos.

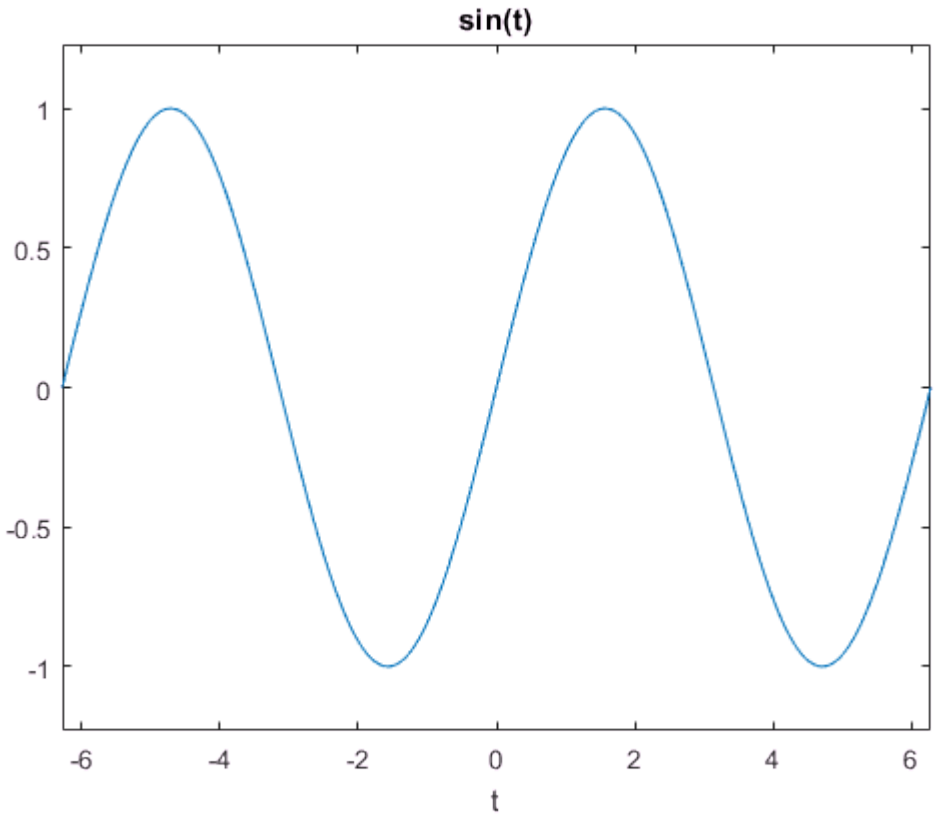
O modo mais simples é utilizar o comando @. Por exemplo, definimos a funcao anonima

f=@(x)(x^.2+2*x+1) define a função dada por $f(x) = x^2 + 2 + 1$.

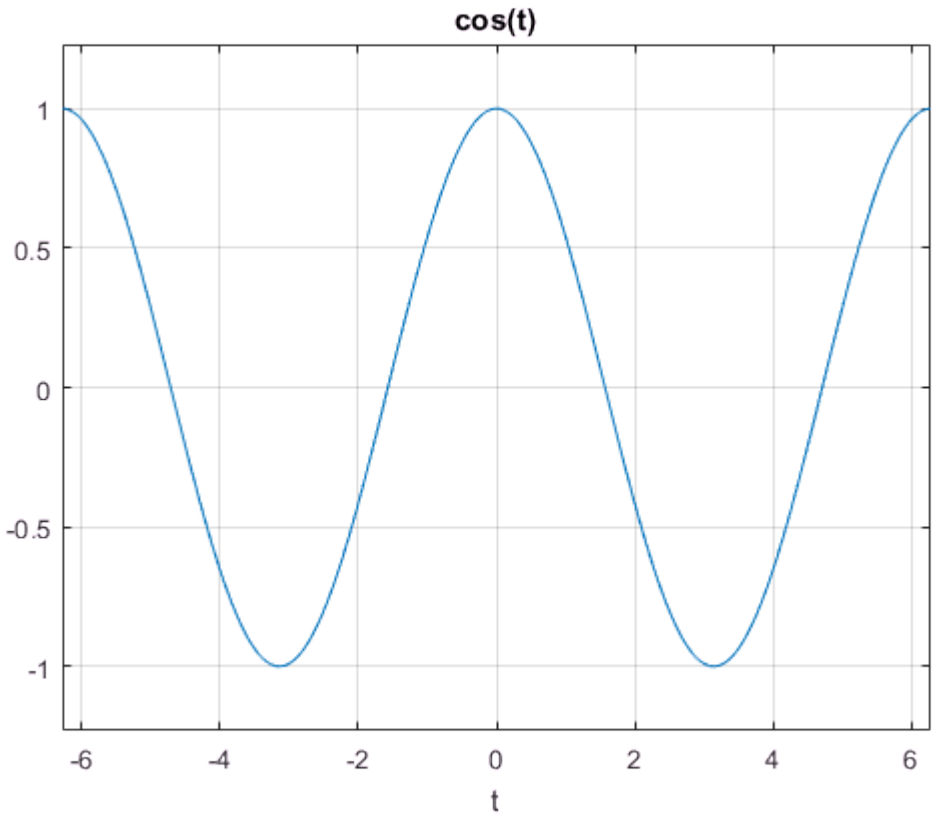
Gráficos

O jeito mais simples de fazer gráfico é usar o comando ezplot

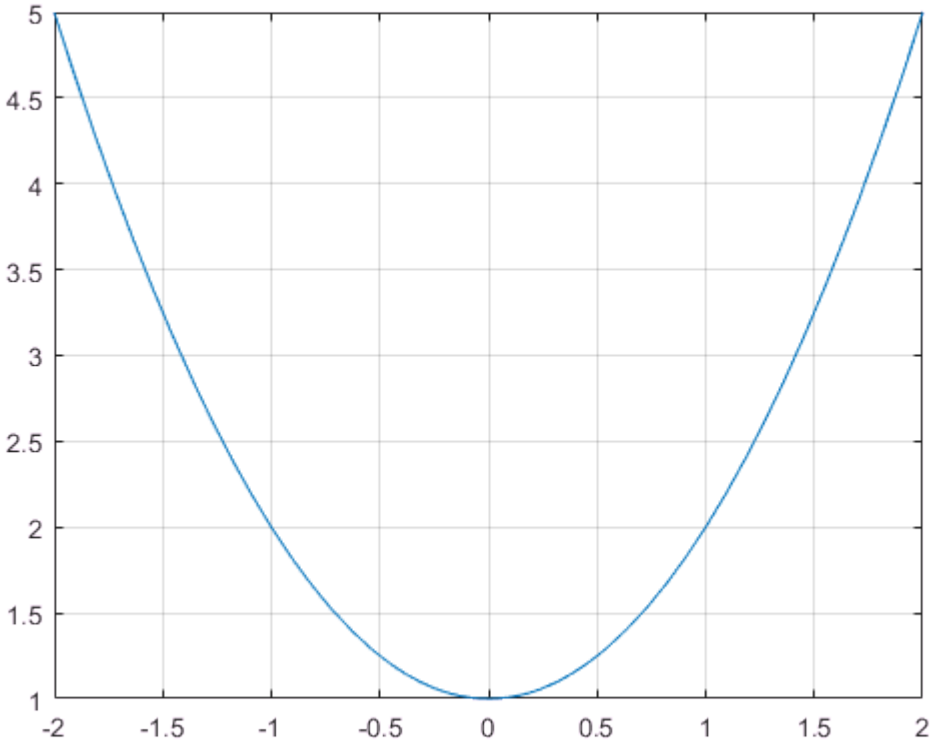
```
ezplot sin(t)
```




```
ezplot cos(t),grid
```

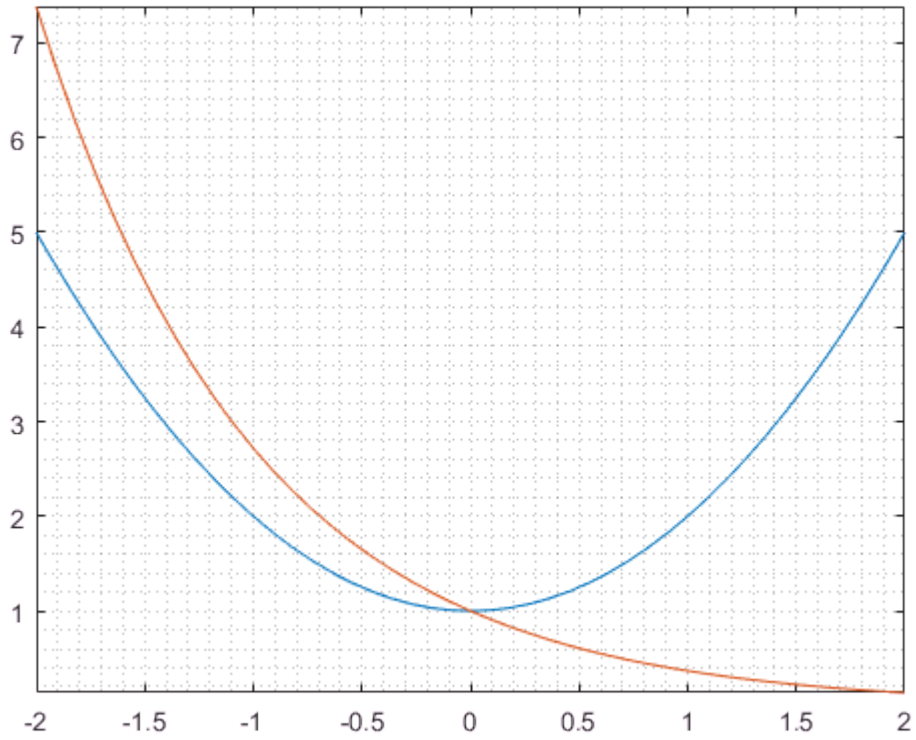



```
%%%%comando fplot
fplot(@(x)(x.^2+1), [-2, 2]),grid
```



Dois gráficos juntos

```
fplot(@(x)( x.^2+1),[-2,2])
hold on
fplot(@(x) exp(-x),[-2,2])
hold off,grid minor
```

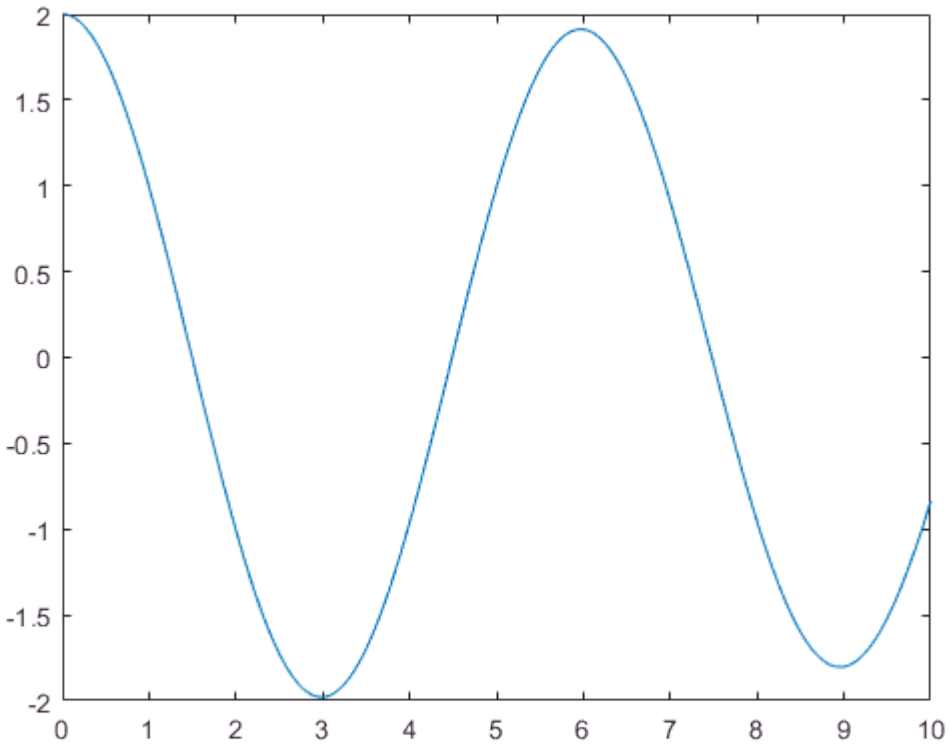


Outro modo de definir uma funçào é utilizar o comando inline. Veja o exemplo.

```
g=inline('cos(x)+cos(1.1*x)')
```

```
g =

     Inline function:
     g(x) = cos(x)+cos(1.1*x)
```

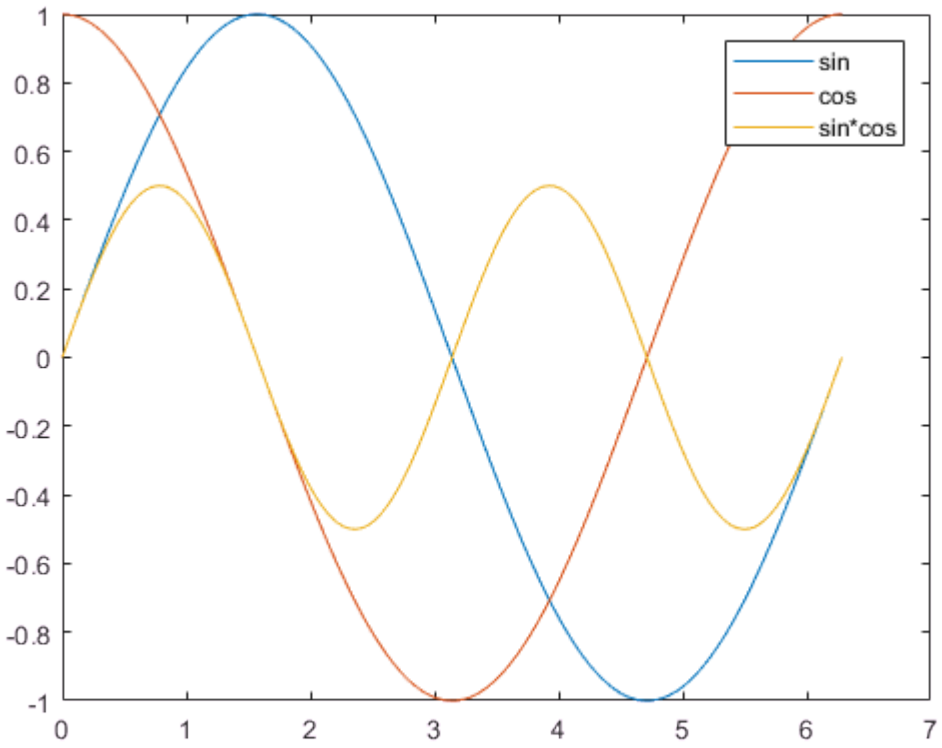
```
x=0:0.01:10;
y=g(x);
plot(x,y)
```



Outro modo de definir uma função é usando o comando syms

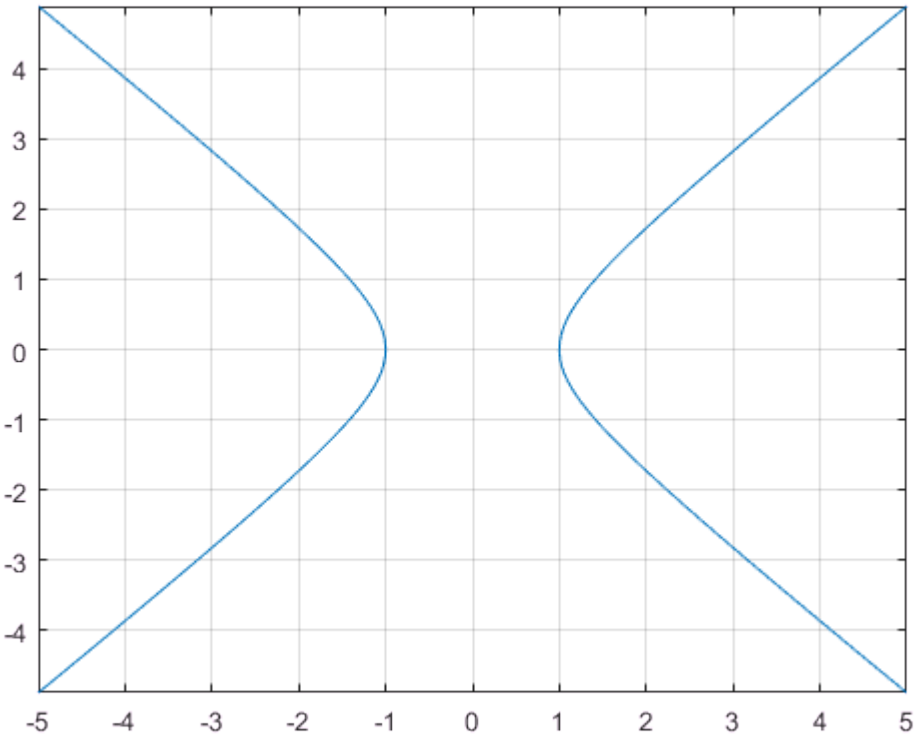
```
syms f(x)
f(x) = sin(x^2);
```

```
 w=0:pi/100:2*pi;
x1= sin(w);
x2= sin(w+pi/2);
x3= x1.*x2;
plot(w,x1,w,x2,w,x3)
legend('sin','cos','sin*cos')
```

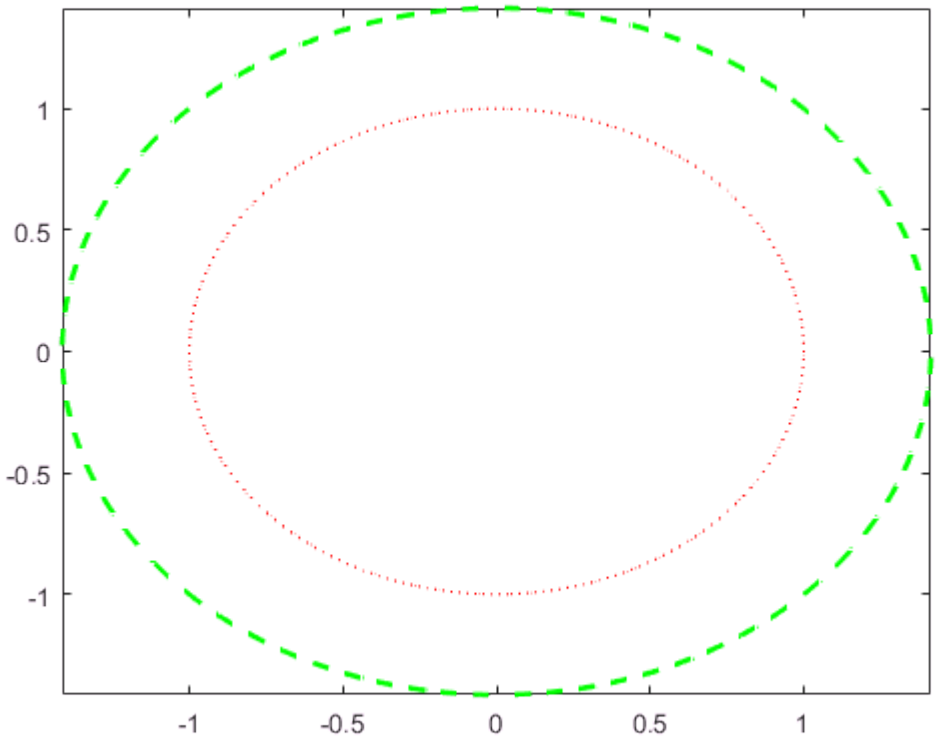


Implicti Plot : veja como desenhar o gráfico de uma função dada implicitamente. $x^2 - y^2 - 1 = 0$

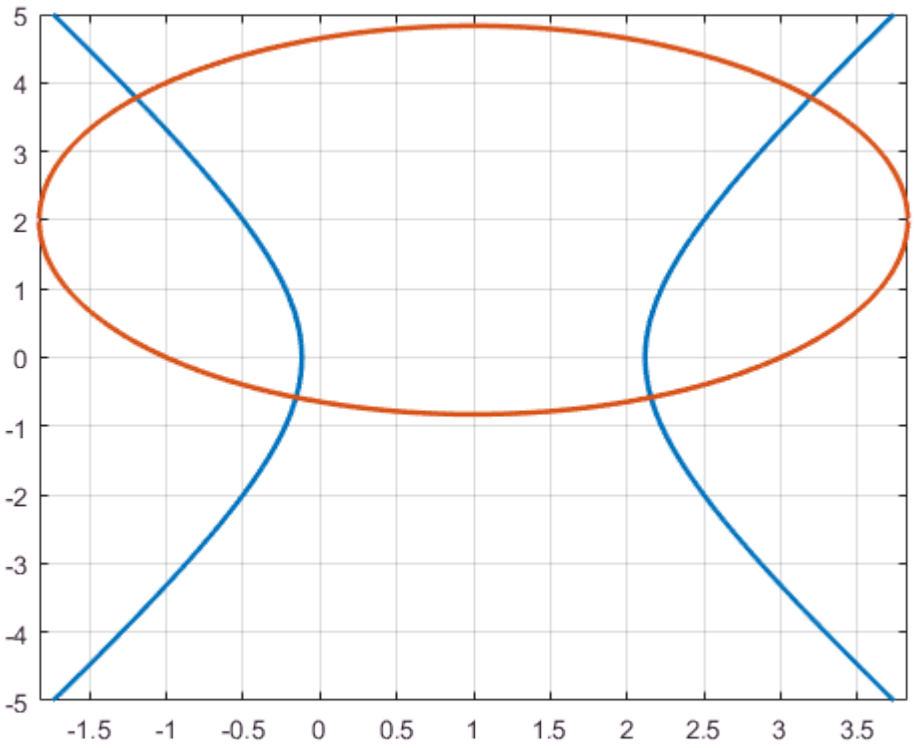
```
fimplicit(@(x,y) x.^2 - y.^2 - 1),grid
```



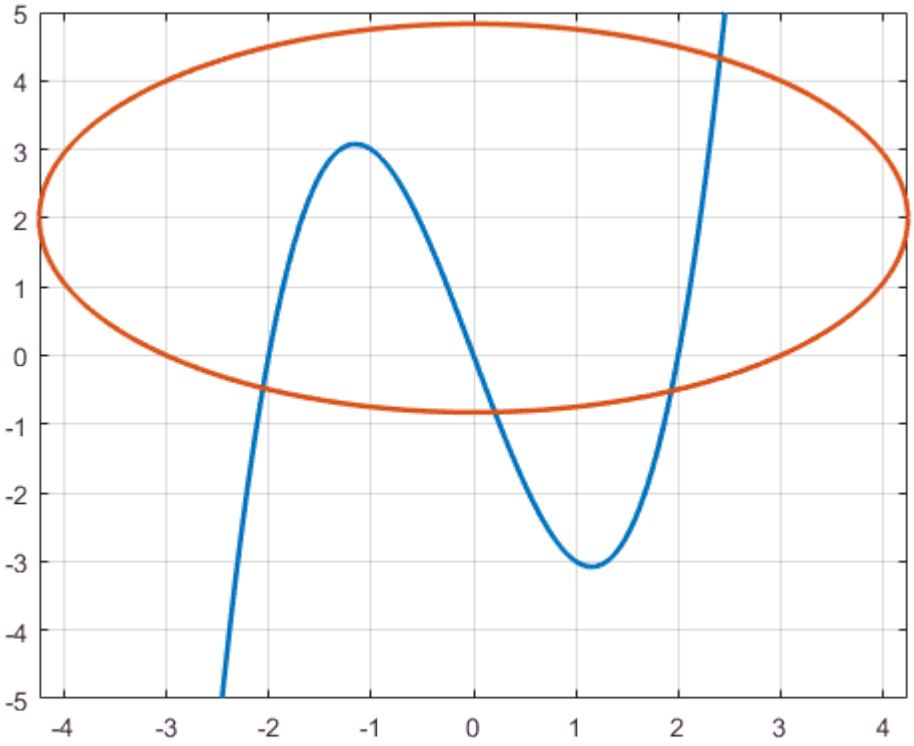

```
f1 = @(x,y) x.^2 + y.^2 - 1;
fimplicit(f1,':r')
hold on
f2 = @(x,y) x.^2 + y.^2 - 2;
fimplicit(f2,'--g','LineWidth',2)
hold off
```



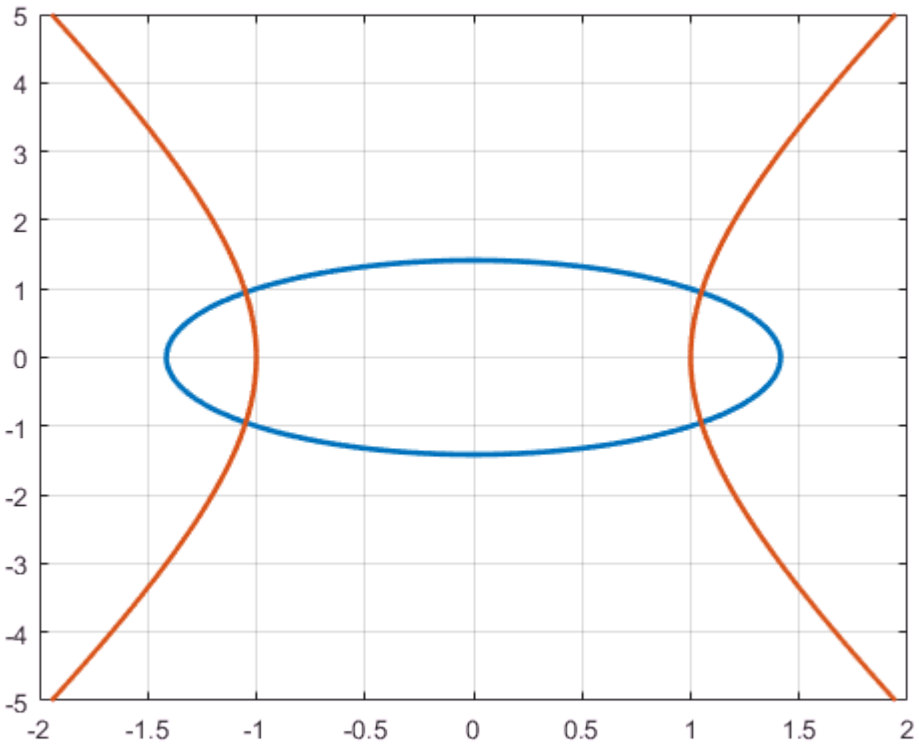

```
f1 = @(x,y) (8*x-4*x.^2 + y.^2 +1)/8;
fimplicit(f1,'LineWidth',2)
hold on
f2 = @(x,y) (2*x-x.^2 +4*y- y.^2 +3)/4;
fimplicit(f2,'LineWidth',2)
hold off,grid,
```




```
f1 = @(x,y) (4*x- x.^3 + y );
fimplicit(f1,'LineWidth',2)
hold on
f2 = @(x,y) (-1*x.^2/9 +(4*y- y.^2+4)/4);
fimplicit(f2,'LineWidth',2)
hold off,grid,
```

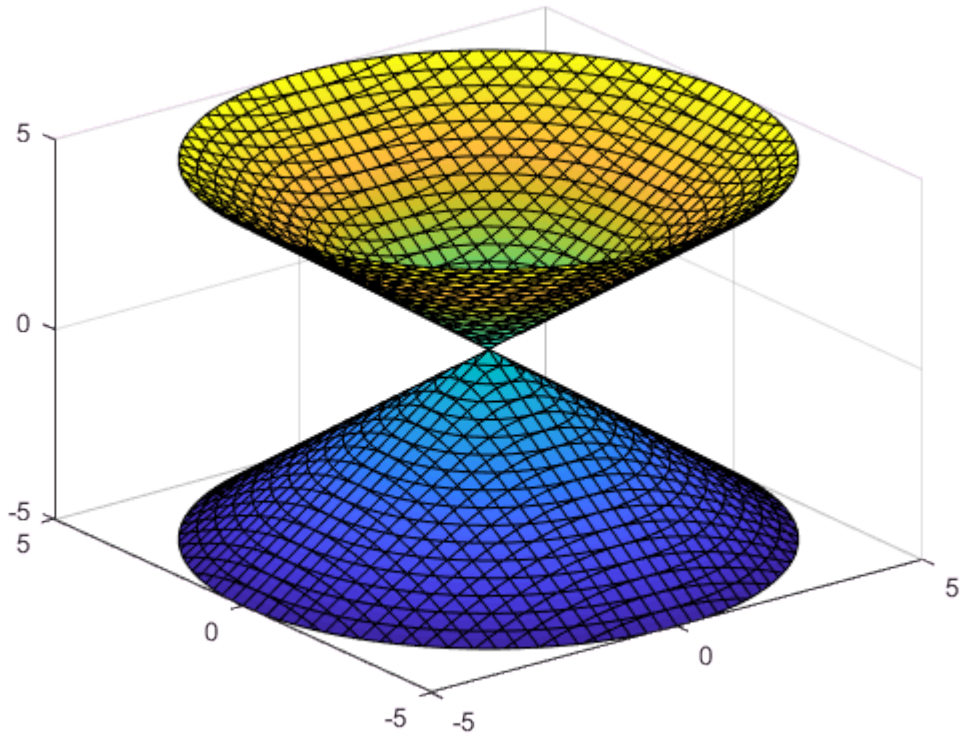



```
f1 = @(x,y) (x.^2 + y.^2-2 );
fimplicit(f1,'LineWidth',2)
hold on
f2 = @(x,y) (x.^2- y.^2/9-1);
fimplicit(f2,'LineWidth',2)
hold off,grid,
```

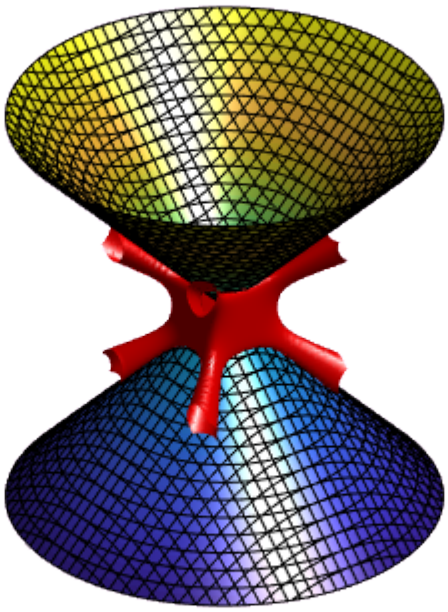


Implicit plot em 3 dimensões

```
f = @(x,y,z) x.^2 + y.^2 - z.^2;
fimplicit3(f)
```




```
fun=@(x,y,z)(1-x.^8-3.*y.^8-2.*z.^8+5.*x.^4.*z.^2.*y.^2+3.*y.^4.*x.^2.*z.^2) ;

[X,Y,Z]=meshgrid(-2:0.1:2,-2:0.1:2,-2:0.1:2);

val=fun(X,Y,Z);

fv=isosurface(X,Y,Z,val,0);

p = patch(fv);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'red');
set(p,'EdgeColor' , 'none');
daspect([1,1,1])
view(3); axis tight
camlight
lighting phong,axis off
```



Som no MatLab

```
r= 2*(rand(1,16000)-0.5)
```

```
r =
    0.6294    0.8116   -0.7460    0.8268    0.2647   -0.8049   -0.4430    0.0938 ···
```

```
sound(r,8000)
```

Se quisermos reproduzir uma sinusóide

de 1000Hz com a duração de 0.5 segundos, utilizando uma frequência de amostragem de 8000Hz

procede-se da seguinte forma

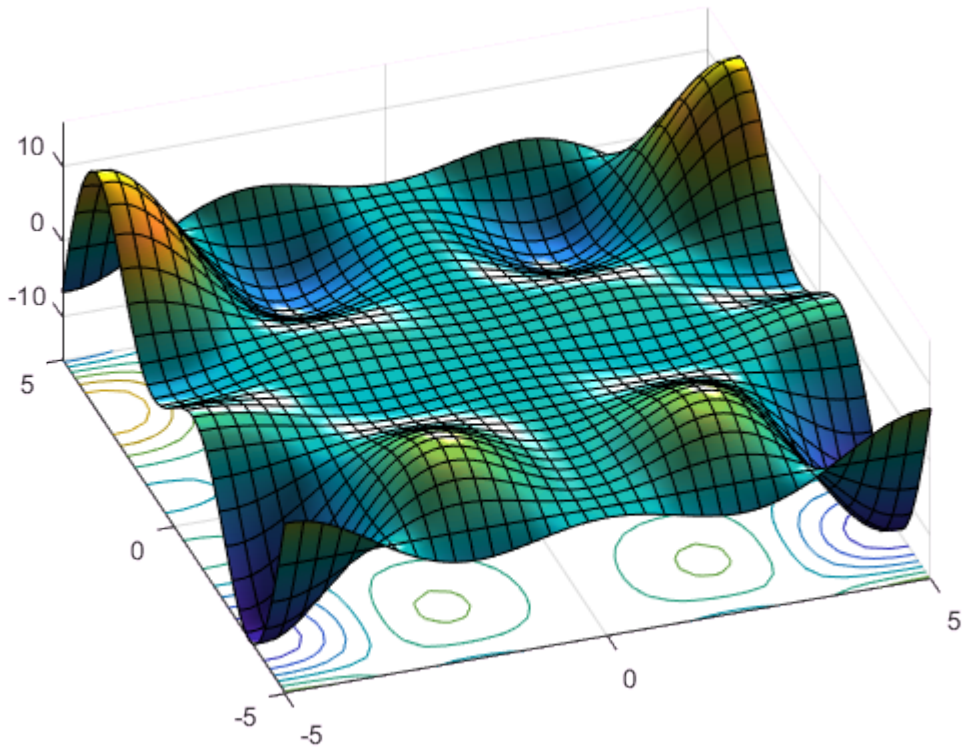
```
fs= 8000;     % Frequência de amostragem
Ts= 1/fs;     % Período de amostragem
t= 0:Ts:0.5;  % Instantes de amostragem
x= sin(2*pi*1000*t);
sound(x,fs)
```

```
fs= 8000;     % Frequência de amostragem
Ts= 1/fs;     % Período de amostragem
t= 0:Ts:0.5;  % Instantes de amostragem
x= sqrt(2*pi*1000*t);
```

```
sound(x,fs)
```

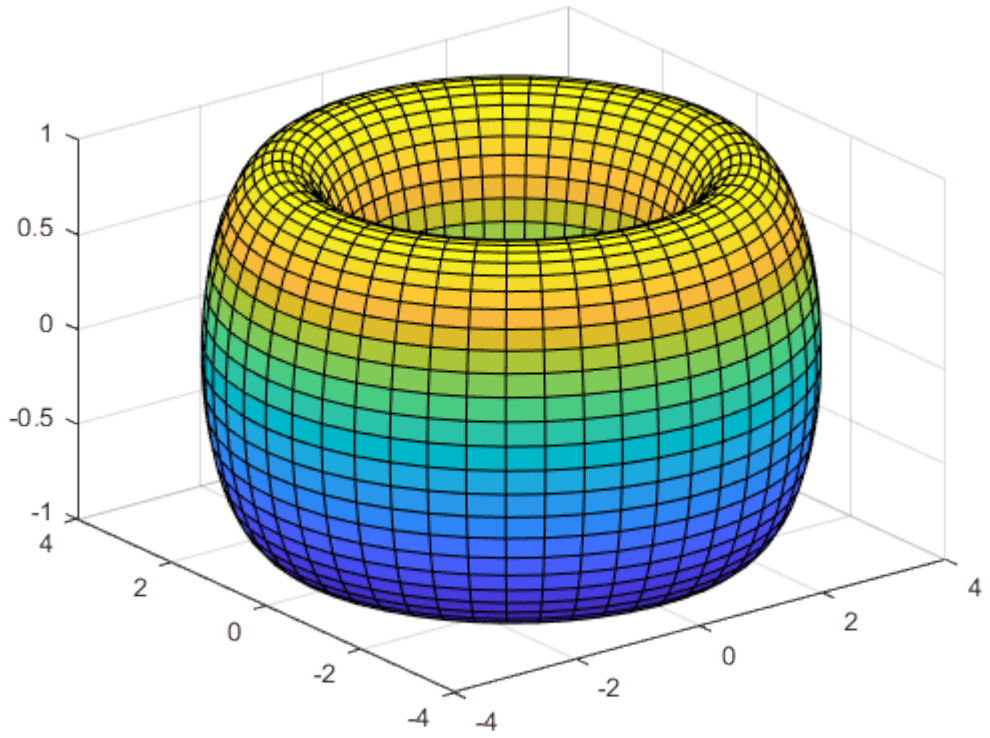
Superficies

```
syms x y
fsurf(x*y*sin(x)*cos(y),'ShowContours','on')
set(camlight,'Color',[0.5 0.5 1]);
set(camlight('left'),'Color', [1 0.6 0.6]);
set(camlight('left'),'Color', [1 0.6 0.6]);
set(camlight('right'),'Color', [0.8 0.8 0.6]);
material shiny
view(-19,56)
```



Superficie parametrizada

```
u=linspace(-pi,pi,50);
v=linspace(-pi,pi,50);
[u,v]=meshgrid(u,v);
x = cos(u).*(cos(v)+3);
y=sin(u).*(cos(v)+3);
z=sin(v);
surf(x,y,z);
```




```
%view([-30.70 73.99]) %uma nova visão
```

Derivada de função

```
syms f(x)
f(x) = sin(x^2);
df = diff(f,x)
```

df(x) = $2x\cos\left(x^2\right)$

```
df(0)
```

ans = 0

```
%%derivada de ordem superior
syms t
d4 = diff(t^6,4)
```

d4 = $360\,t^2$

Integral

```
syms x
int(x*sin(x),x)
```

ans = $\sin(x) - x\cos(x)$

```
syms x
int(x*sin(x),x,0,pi/2)
```

ans = 1

Raízes de polinômios: $p(x) = x^4 + 10x^3 + 35x^2 + 50x + 24$

```
p=[1 10 35 50 24]
```

```
p =
     1    10    35    50    24
```

```
raizes=roots(p)
```

```
raizes =
   -4.0000
   -3.0000
   -2.0000
   -1.0000
```

Zeros de funções: resolver $f(x) = 0$. Precisa de um chute inicial.

```
parabola = @(x) x.^2-9;
x = fzero(parabola,2)
```

x = 3

```
%%OU
fsolve(parabola,2)
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.
```

```
<stopping criteria details>
ans = 3
```

Este exemplo eu conheço!

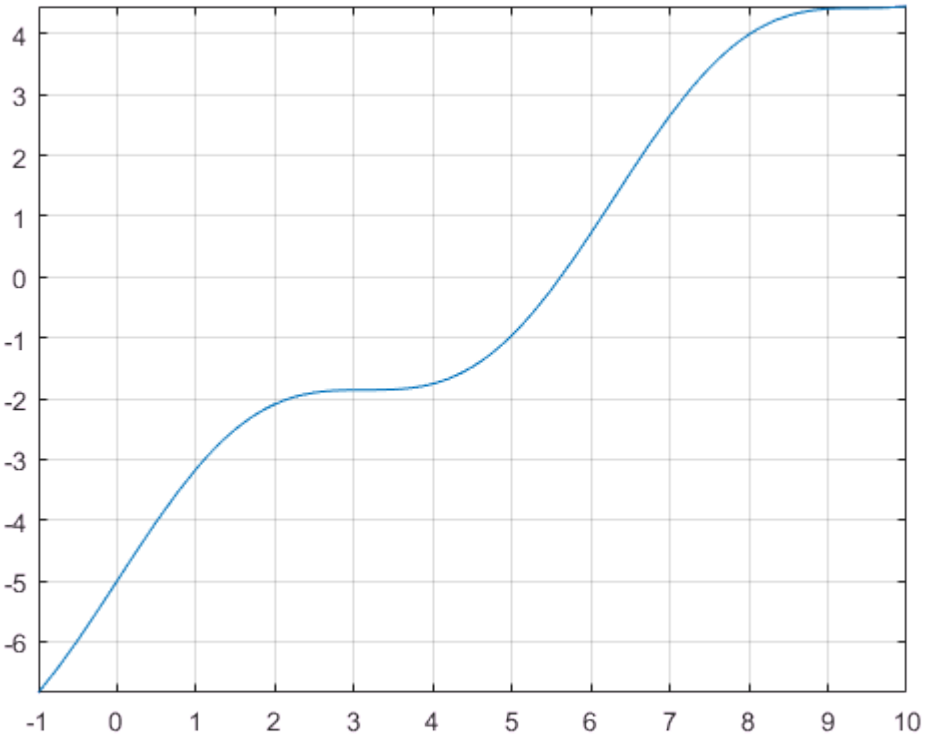
```
f = @(x)(x-cos(x));
x = fsolve(f,0.70)
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
x = 0.7391
```

```
solve(@(x)sin(x)+x-5)
```

```
Warning: Cannot solve symbolically. Returning a numeric approximation instead.
ans = 5.617555005272698917621392157 1114
```

```
fplot(@(x)(sin(x)+x-5), [-1, 10]),grid
```



Exemplo de fsolve para sistema de equações nao lineares.

$$y_1 = 3x_1^2 + 4x_2^2 - 16$$
$$y_2 = 2x_1^2 - 3x_2^2 - 5$$

```
fsolve(@(x)[3*x(1).^2+4*x(2).^2-16;2*x(1).^2-3*x(2).^2-5],[1,1,1])
```

```
Warning: Trust-region-dogleg algorithm of FSOLVE cannot handle non-square systems; using
Levenberg-Marquardt algorithm instead.
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
ans =
    2.0000    1.0000    1.0000
```

Usando fsolve para resolver F(x)=0 onde

$$F(x) = (x_1^2 + x_2x_3, x_1 + 2x_2 - 3x_3, \sin(x_1 + 2x_2 - 3x_x)$$

```
fsolve(@(x)[x(1).^2+x(2).*x(3); x(1)+2*x(2)-3*x(3);sin(x(1)-x(2)+x(3))],[1,1,1])
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
ans =
    0.0021    0.0084    0.0063
```

```
```

Sistemas lineares: resolver Ax=b

```
A=[1 -2 -3; 1 2 -1; 2 4 -1]
```

```
A =
     1    -2    -3
     1     2    -1
     2     4    -1
```

```
b=[1; 2 ;3]
```

```
b =
     1
     2
     3
```

```
x= A\b
```

```
x =
   -0.5000
    0.7500
   -1.0000
```

```

```

Usando o comando linsolve

```
x = linsolve(A,b)
```

```
x =
   -0.5000
    0.7500
   -1.0000
```

-

Jacobiana

```
syms x y z
jacobian([x*y*z, y^2, x + z], [x, y, z])
```

```
ans =
```

$$\begin{pmatrix} yz & xz & xy \\ 0 & 2y & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

```
syms x y
 f=[sin(x*y); x^2+y^2; 3*x-2*y]
```

```
f =
```

$$\begin{pmatrix} \sin(xy) \\ x^2 + y^2 \\ 3x - 2y \end{pmatrix}$$

```
 Jf=jacobian(f)
```

```
Jf =
```

$$\begin{pmatrix} y\cos(x\,y) & x\cos(x\,y) \\ 2\,x & 2\,y \\ 3 & -2 \end{pmatrix}$$

Rotacional

```
syms x y z
curl([x^3*y^2*z, y^3*z^2*x, z^3*x^2*y], [x, y, z])
```

```
ans =
```

$$\begin{pmatrix} x^2 z^3 - 2\,x\,y^3 z \\ x^3 y^2 - 2\,x\,y\,z^3 \\ y^3 z^2 - 2\,x^3\,y\,z \end{pmatrix}$$

Gradiente

```
syms x y z
f = 2*y*z*sin(x) + 3*x*sin(z)*cos(y);
gradient(f, [x, y, z])
```

```
ans =
```

$$\begin{pmatrix} 3\cos(y)\sin(z) + 2\,y\,z\cos(x) \\ 2\,z\sin(x) - 3\,x\sin(y)\sin(z) \\ 2\,y\sin(x) + 3\,x\cos(y)\cos(z) \end{pmatrix}$$

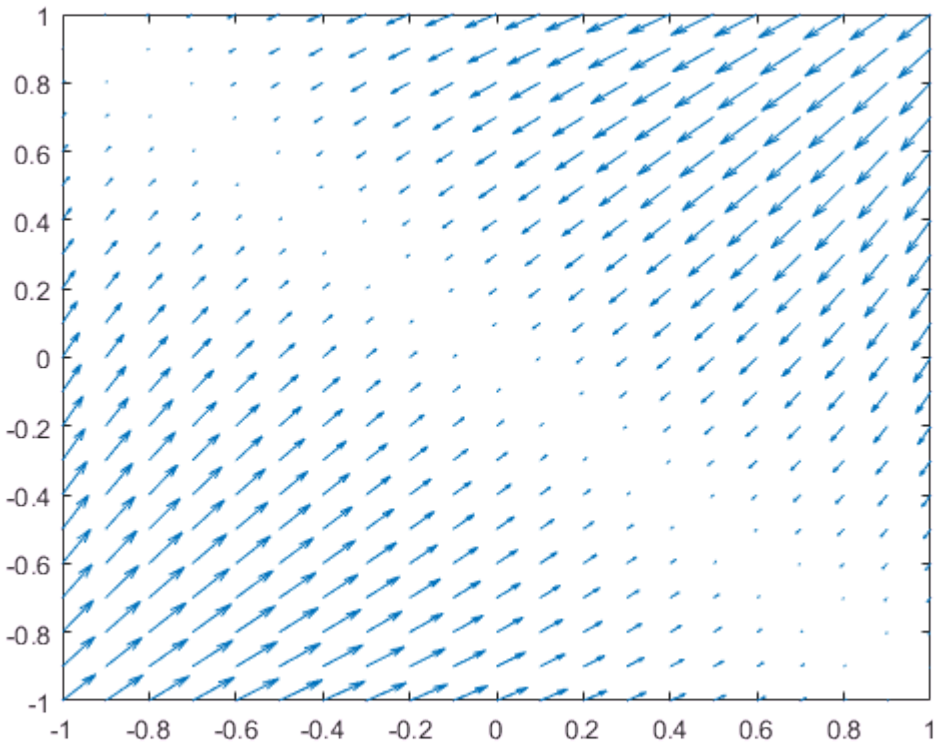
Plotando o gradiente de f

```
syms x y
f = -(sin(x) + sin(y))^2;
g = gradient(f, [x, y])
```

```
g =
```

$$\begin{pmatrix} -2\cos(x) & (\sin(x) + \sin(y)) \\ -2\cos(y) & (\sin(x) + \sin(y)) \end{pmatrix}$$

```
%%agora plotando o campo gradiente
[X, Y] = meshgrid(-1:.1:1,-1:.1:1);
G1 = subs(g(1), [x y], {X,Y});
G2 = subs(g(2), [x y], {X,Y});
quiver(X, Y, G1, G2)
```



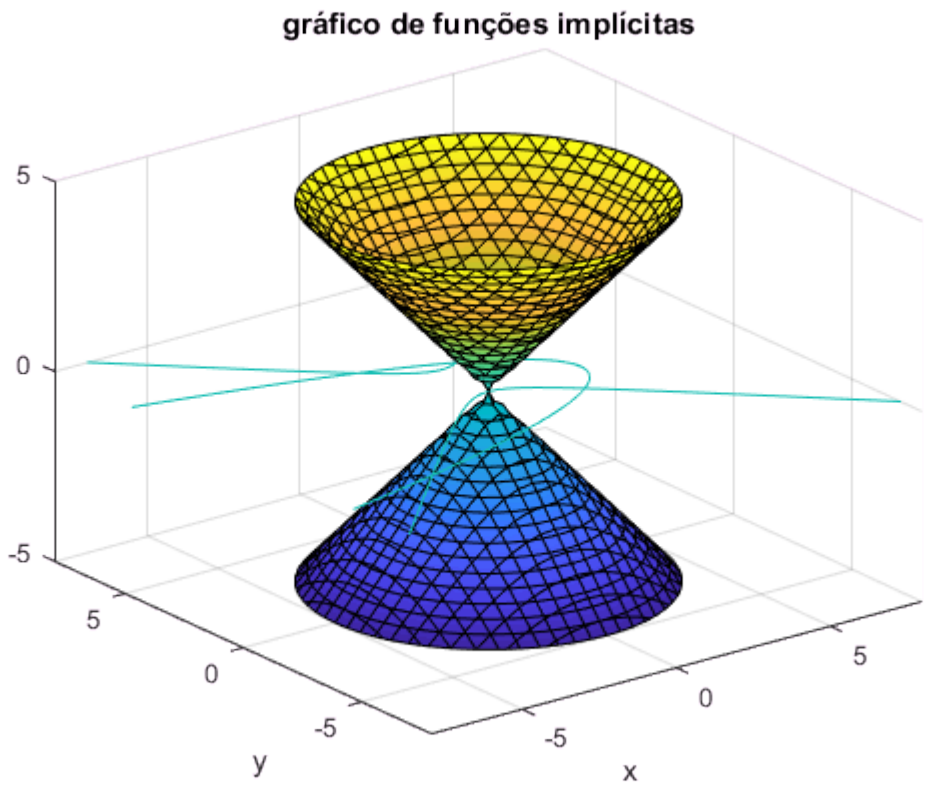
Hessiana

```
syms x y z
f = x*y + 2*z*x;
hessian(f,[x,y,z])
```

```
ans =
```

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix}$$

```
hold on
 ezplot('x^2-y^2+2*y',[-8,8,-8,8])
 ezplot('2*x+y^2-6',[-8,8,-8,8])
 hold off
 title('gráfico de funções implícitas')
 grid on
```




```
f=@(x)( -x.^2+14*x+21), q=integral(f,0,10)
```

```
f = function_handle with value:
    @(x)(-x.^2+14*x+21)
q = 576.6667
```

Calcular a integral dupla

$$\int_1^2 \int_0^3 x^2 y dy dx$$

```
 f=@(x,y)( x.^2.*y);
res=integral2(f,1,2,0,3);
fprintf('O valor da integra dupla é: %2.3f\n',res)
```

```
O valor da integra dupla é: 10.500
```

Programação

Estrutura if

if condição

declaraçoes

end

Estrutura if ...else

if condiçao

declaracaoes

else

declaracoes

end

Estrutura if ..elseif

if condicao

declaracao

elseif condicao

declaracoes

....

else

declaracoes(do elese)

end

Laços

for indice= inicio:passo:fim

declaracoes

end

While

while condicao

declaracao

end

Máximos e Mínimos

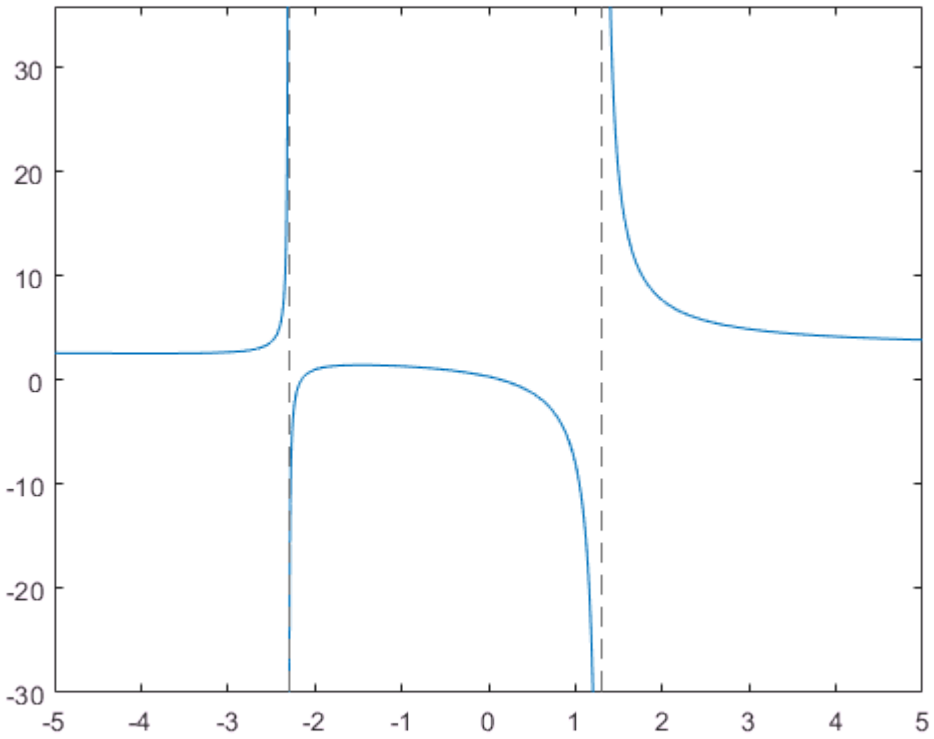
Estudando max e min da função $f(x) = \frac{3x^2 + 6x - 1}{x^2 + x - 3}$

```
syms x
num = 3*x^2 + 6*x -1;
denom = x^2 + x - 3;
f = num/denom,
```

```
f =
```

$$\frac{3\,x^2 + 6\,x - 1}{x^2 + x - 3}$$

```
fplot(f),
```




```
roots = solve(denom)
```

```
roots =
```

$$\begin{pmatrix} -\frac{\sqrt{13}}{2} - \frac{1}{2} \\ \frac{\sqrt{13}}{2} - \frac{1}{2} \end{pmatrix}$$

```
f1 = diff(f)
```

f1 =

$$\frac{6x+6}{x^2+x-3} - \frac{(2x+1)\,(3x^2+6x-1)}{(x^2+x-3)^2}$$

```
f1 = simplify(f1)
```

f1 =

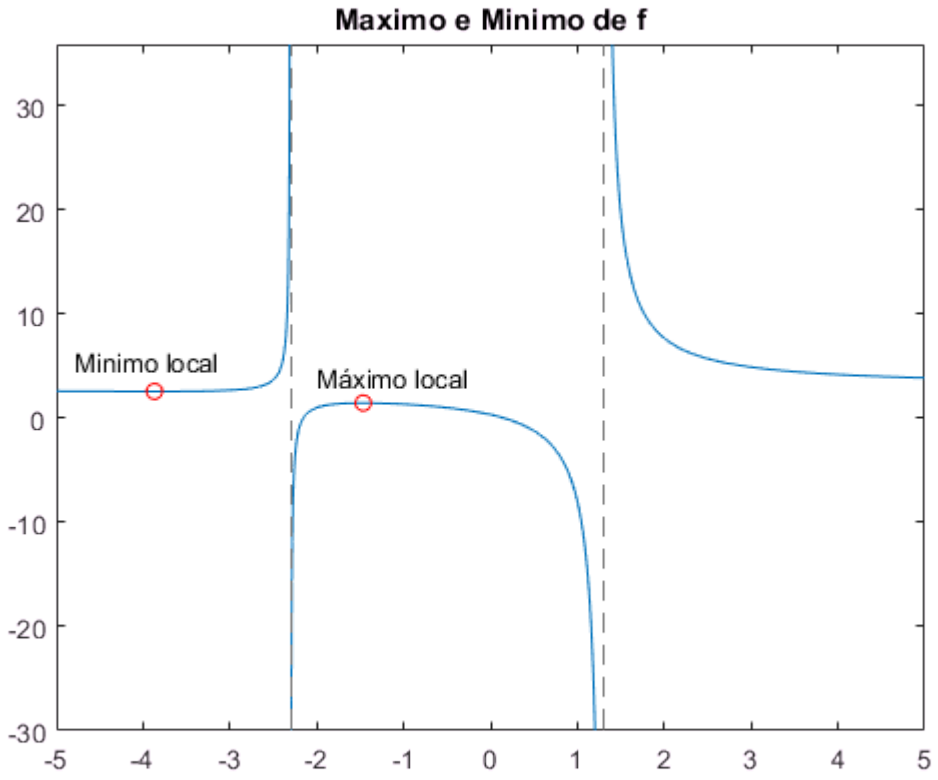
$$-\frac{3x^2+16x+17}{(x^2+x-3)^2}$$

```
crit_pts = solve(f1)
```

crit_pts =

$$\begin{pmatrix} -\frac{\sqrt{13}}{3} - \frac{8}{3} \\ \frac{\sqrt{13}}{3} - \frac{8}{3} \end{pmatrix}$$

```
fplot(f)
hold on
plot(double(crit_pts), double(subs(f,crit_pts)),'ro')
title('Maximo e Minimo de f')
text(-4.8,5.5,'Minimo local')
text(-2,4,'Máximo local')
hold off
```




```
%%inflexão
f2 = diff(f1);
inflec_pt = solve(f2,'MaxDegree',3);
double(inflec_pt)
```

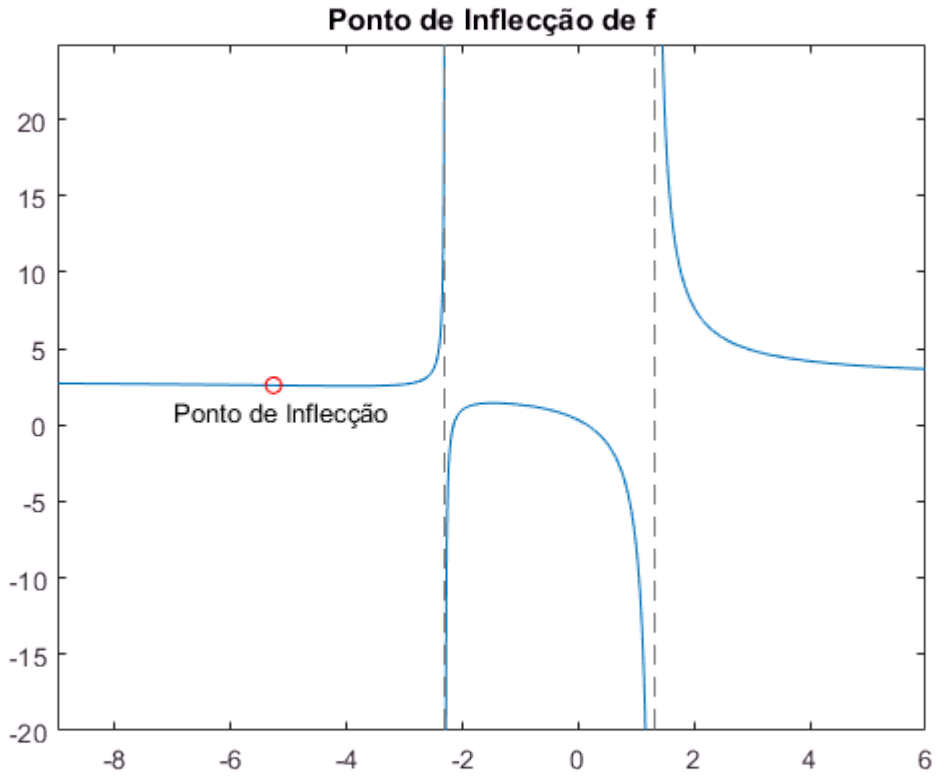
```
ans =
   -5.2635 + 0.0000i
   -1.3682 - 0.8511i
   -1.3682 + 0.8511i
```

```
inflec_pt = inflec_pt(1);
simplify(inflec_pt)
```

```
ans =
```

$$-\frac{2^{2/3}\,13^{1/3}\,\left(13-3\,\sqrt{13}\right)^{1/3}}{6}-\frac{2^{2/3}\,13^{1/3}\,\left(3\,\sqrt{13}+13\right)^{1/3}}{6}-\frac{8}{3}$$

```
fplot(f, [-9 6])
hold on
plot(double(inflec_pt), double(subs(f,inflec_pt)),'ro')
title('Ponto de Inflecção de f')
text(-7,1,'Ponto de Inflecção')
hold off
```



Transformada de Laplace

```
syms a t s
f = exp(-a*t);
F=laplace(f, s),
```

F =

$$\frac{1}{a+s}$$

```
ilaplace(F, s, t) %inversa
```

ans = e^{-at}

Fourier

```
syms x y
f = exp(-x^2);
F=fourier(f, x, y)
```

F =

$$\sqrt{\pi}\, e^{-\frac{y^2}{4}}$$

```
ifourier(F, y, x)
```

ans = e^{-x^2}

Integração numérica

Calcule a integral $\int_0^1 e^{x^2} dx$

```
f=@(x)exp(x.^2)
```

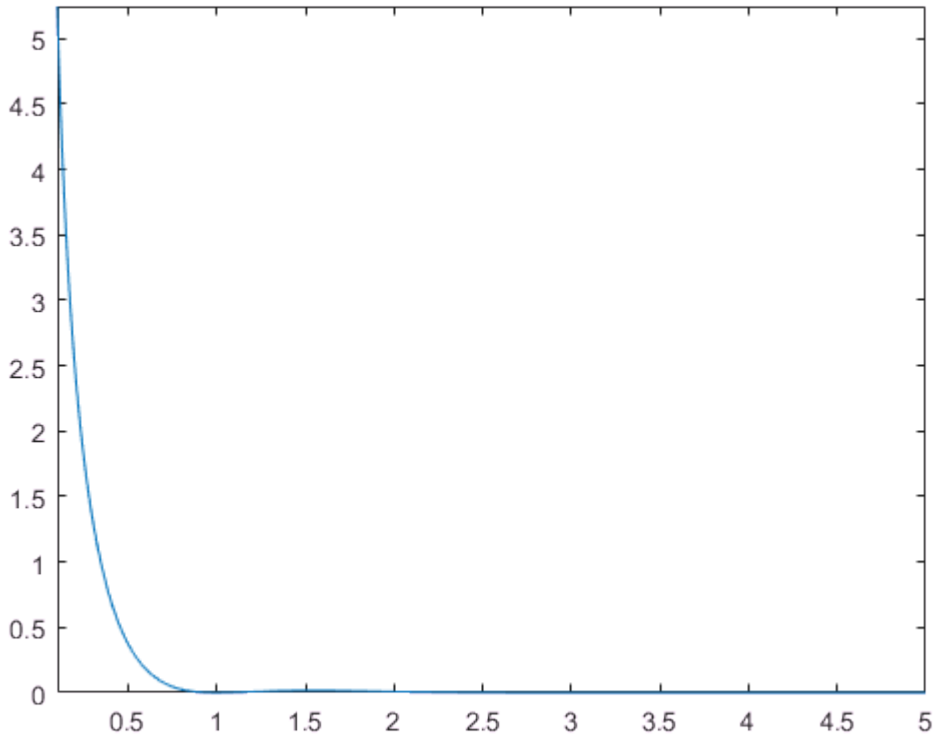
```
f = function_handle with value:
    @(x)exp(x.^2)
```

```
a = integral(f,0,1)
```

```
a = 1.4627
```

Exemplo: calcular a integral $\int_0^\infty e^{-x^2} \ln^2(x) dx$

```
fun = @(x) exp(-x.^2).*log(x).^2;
fplot(fun,[0.1,5])
```




```
q = integral(fun,0,Inf)
```

```
q = 1.9475
```

Usando a regra dos trapézios

```
X = [1 2.5 7 10];
Y = [5.2   7.7   9.6   13.2];
Q1 = trapz(X,Y,2)
```

```
Q1 = 82.8000
```

EDO's

Exemplo: resolver o PVI

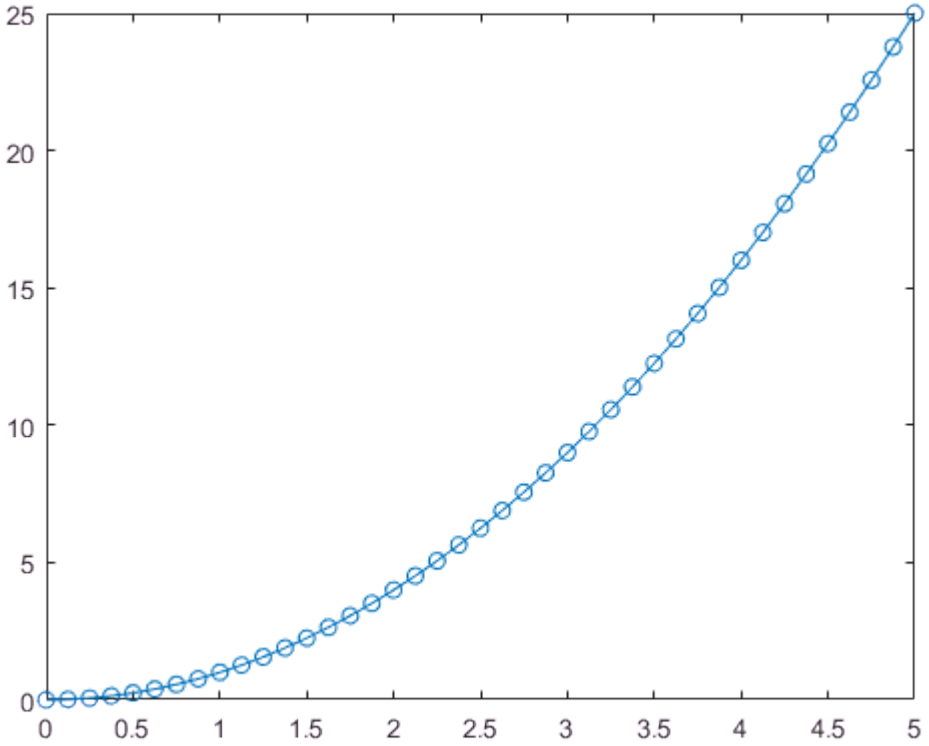
$y' = 2t$
$y(0) = 0$'

no intervalo [0,5].

```
tspan = [0 5];
y0 = 0;
```

```
[t,y] = ode45(@(t,y) 2*t, tspan, y0);
```

```
plot(t,y,'-o')
```



Exemplo: $y' = -2 * y$
$y(0) = 1$

```
tspan = [0 5];
y0 = 1;
[t,y] = ode45(@(t,y) -2*y, tspan, y0),
```

```
t =
         0
    0.0251
    0.0502
    0.0754
    0.1005
    0.2032
    0.3059
    0.4086
    0.5113
    0.6093
     ⋮
```

•

```
y =
    1.0000
    0.9510
    0.9044
```

```
    0.8601
    0.8180
    0.6659
    0.5421
    0.4416
    0.3598
    0.2957
     ⋮
```

```
plot(t,y,'-o')
```