Introduction to MATLAB

5: Graphics with MATLAB

Georgios Georgiou

Department of Mathematics and Statistics

University of Cyprus

Πανεπιστήμιο Κύπρου
Τμήμα Μαθηματικών
και Στατιστικής

plot

plot
title
xlabel
ylabel
legend

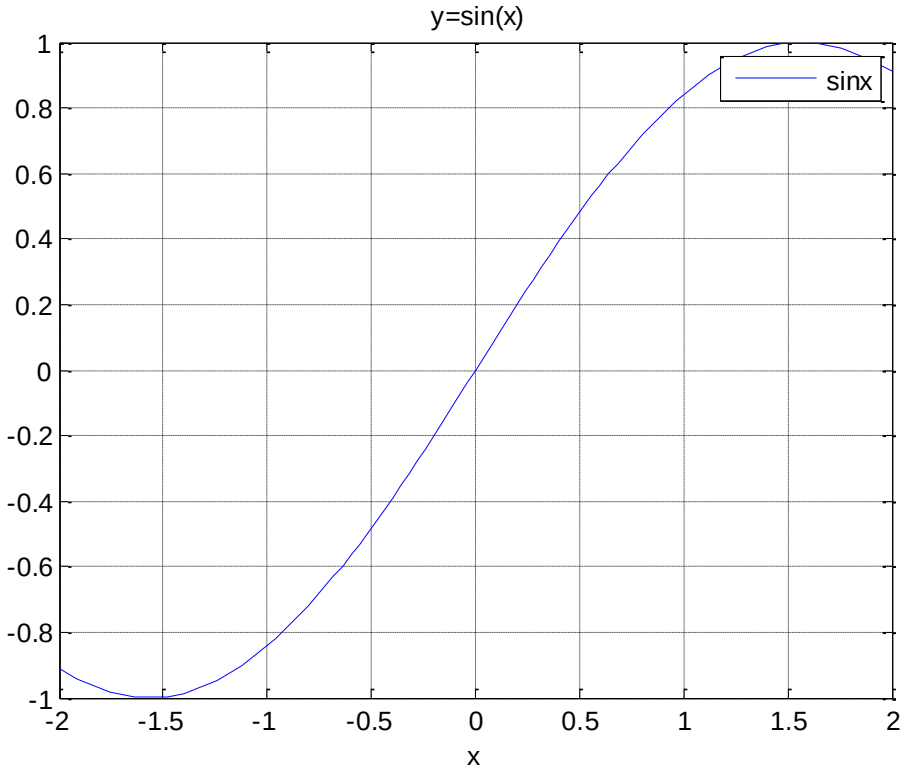
text
grid

figure
plotedit
hold on, hold off

axis

plot

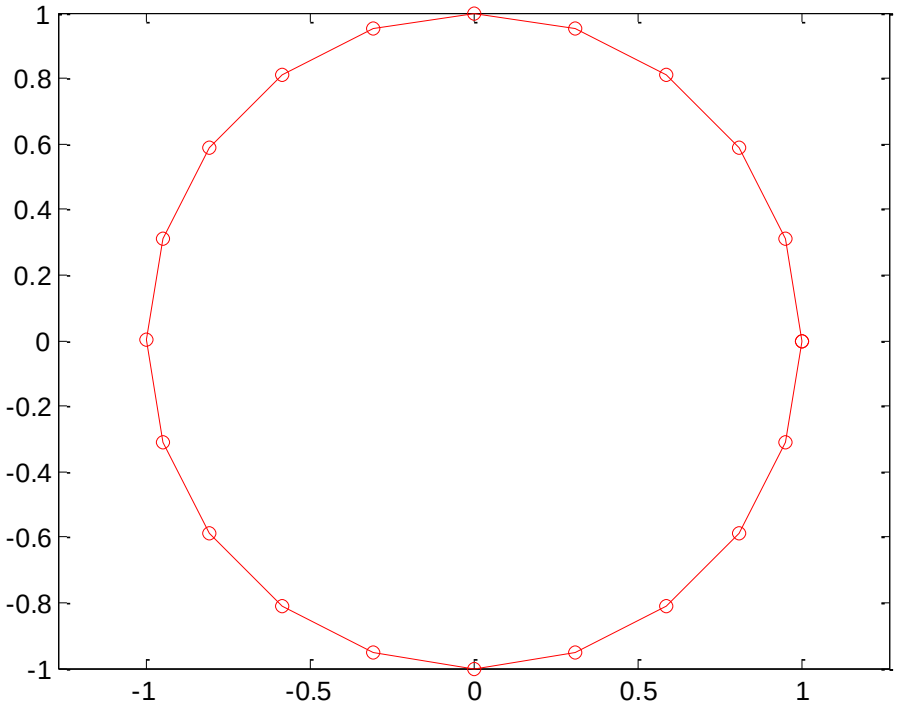
```
>> x=linspace(-2,2,101); plot(x, sin(x))
>> xlabel('x'), title('y=sin(x)')
>> grid
>> legend('sinx')
```



Much more options in the figure window!

Plot with complex functions

```
>> t=0:pi/10:2*pi;
>> plot(exp(i*t),‘r-o')
>> axis equal
```

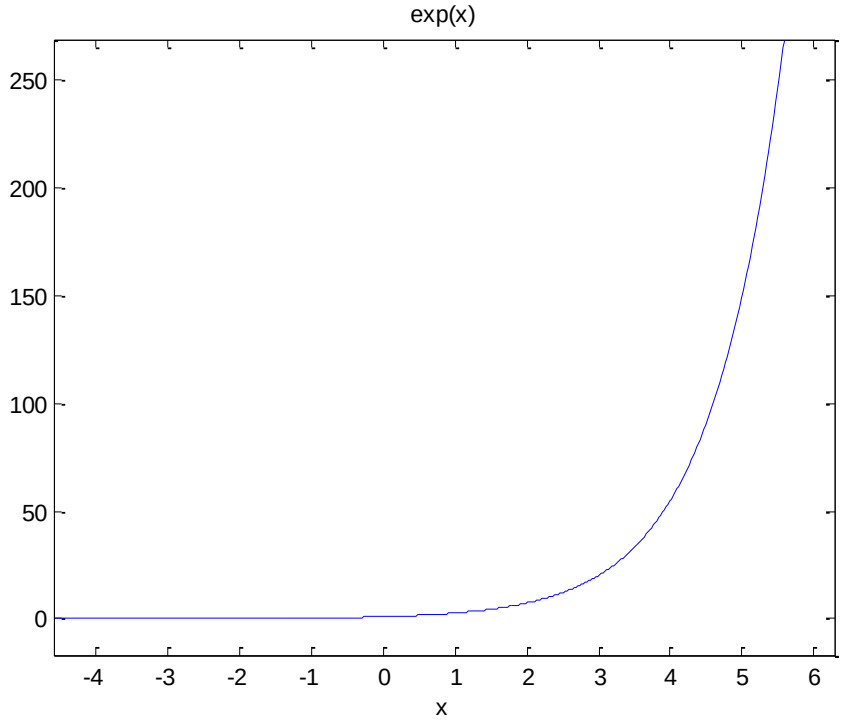


plot(Z) is equivalent to **plot(real(Z), imag(Z))**.

ezplot

- ezplot(f) plots f(x) in [-2π, 2π].
- ezplot(f, a, b) or ezplot(f, [a, b]) plot f(x) in [a,b]
- ezplot(f, [xmin xmax ymin ymax])

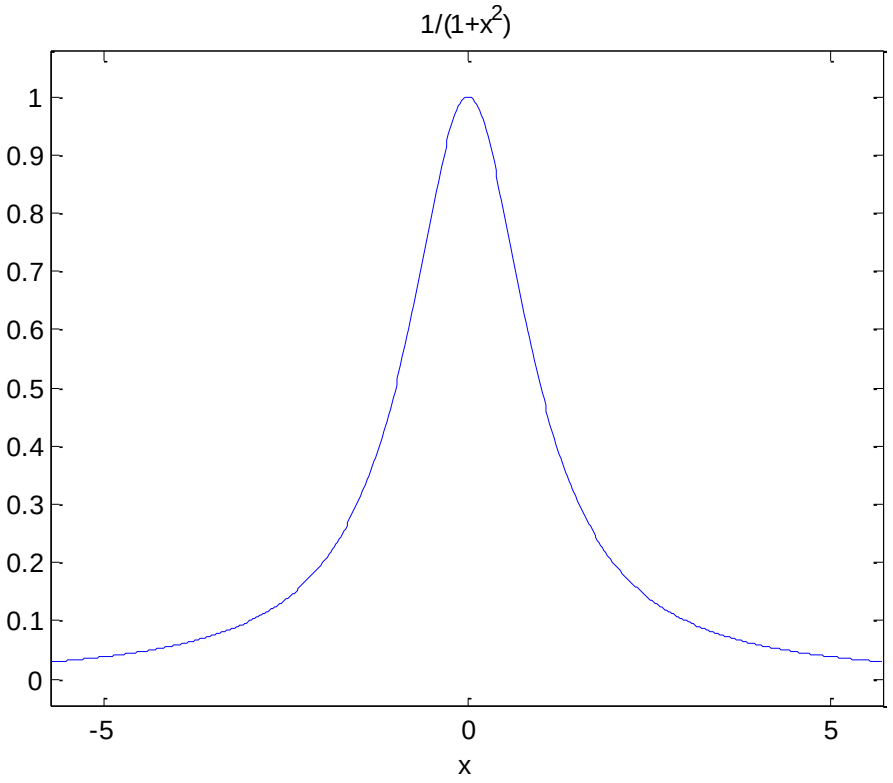
```
>> ezplot('exp(x)')
```



Example 1

$$f(x) = \frac{1}{1+x^2}$$

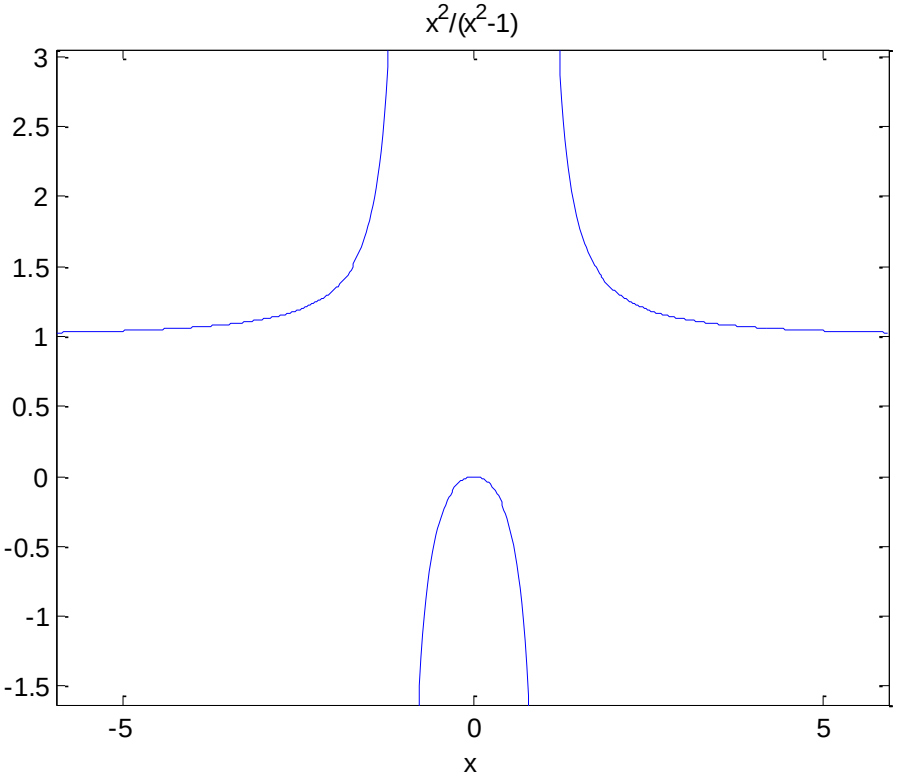
```
>> ezplot('1./(1+x.^2)')
```



Example 2

$$f(x) = \frac{x^2}{x^2 - 1}$$

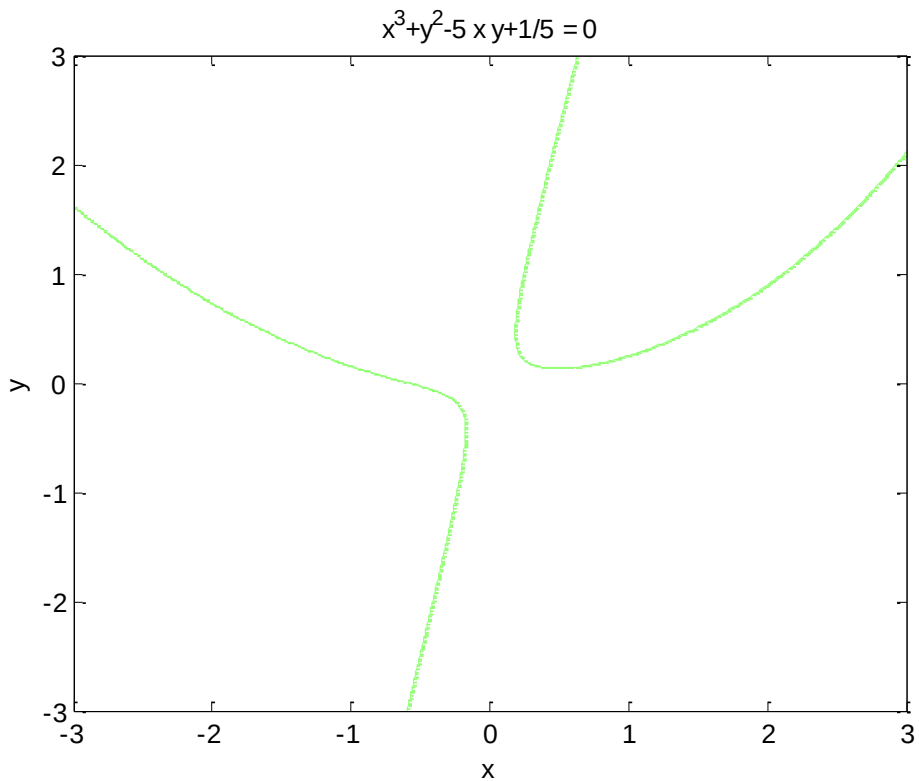
```
>> f = @(x) x.^2./(x.^2-1);
>> ezplot(f)
```



ezplot: implicit functions

$$f(x,y) = x^3 + y^2 - 5xy + \frac{1}{5} = 0$$

```
>> ezplot('x^3+y^2-5*x*y+1/5',[-3,3])
```

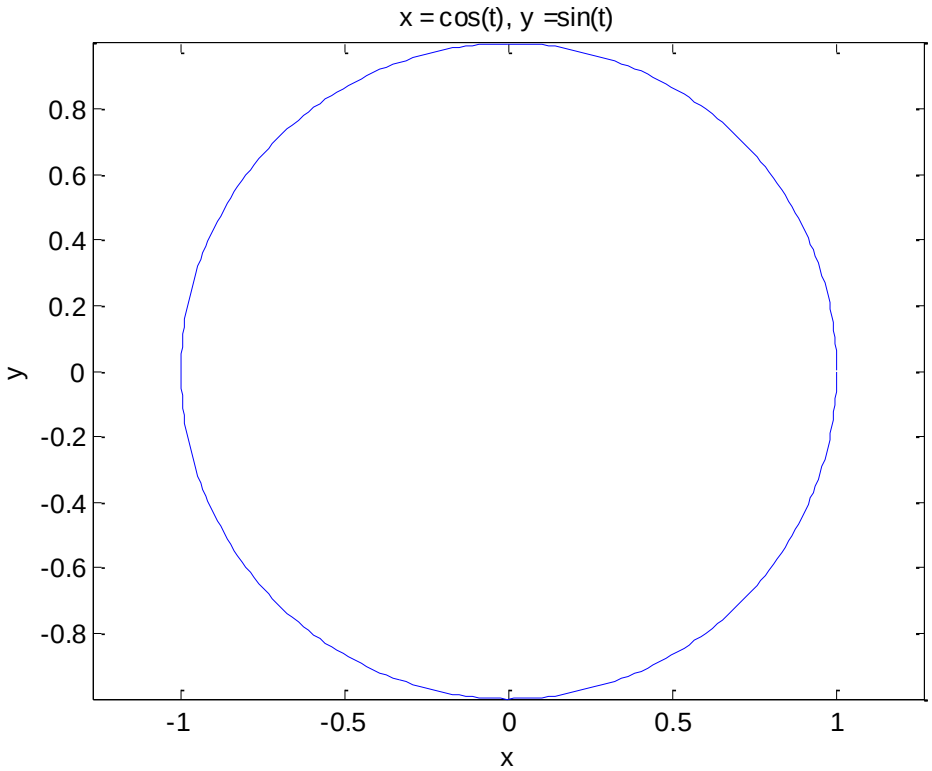


ezplot: parametric curves

$$x = x(t), \quad y = y(t), \quad t \in [a,b]$$

ezplot('x(t)', 'y(t)', [a,b])

```
>> ezplot('cos(t)', 'sin(t)',[0,2*pi])
```

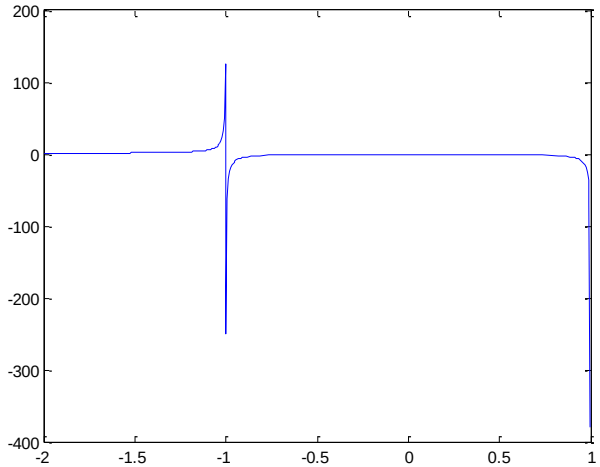


fplot

fplot(f, [xmin, xmax])

fplot(f, [xmin, xmax, ymin, ymax])

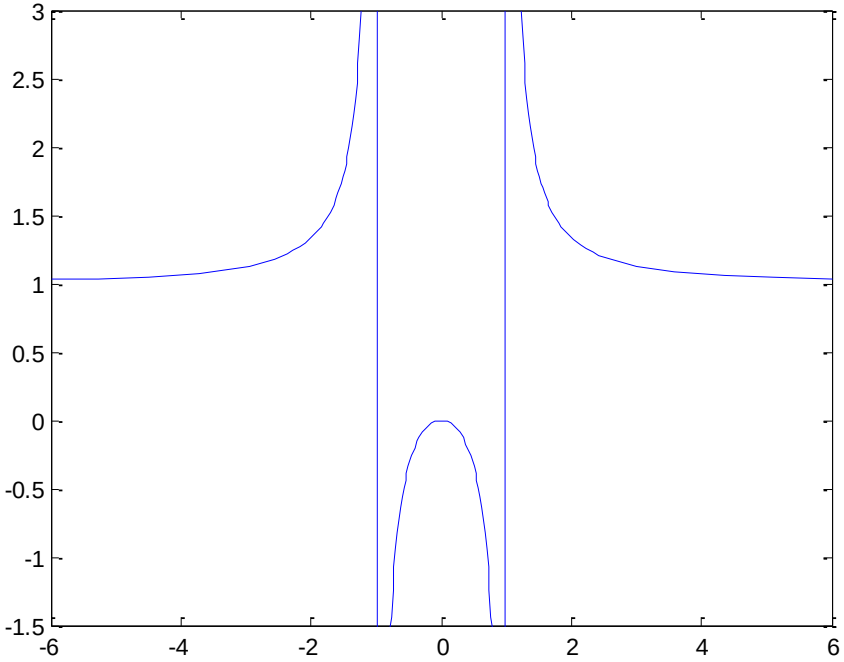
```
>> f = @(x) x.^2./(x.^2-1);
>> fplot(f, [-2,1])
```



$$f(x) = \frac{x^2}{x^2 - 1}$$

fplot

```
>> f = @(x) x.^2./(x.^2-1);
>> fplot(f,[-6 6 -1.5 3])
```

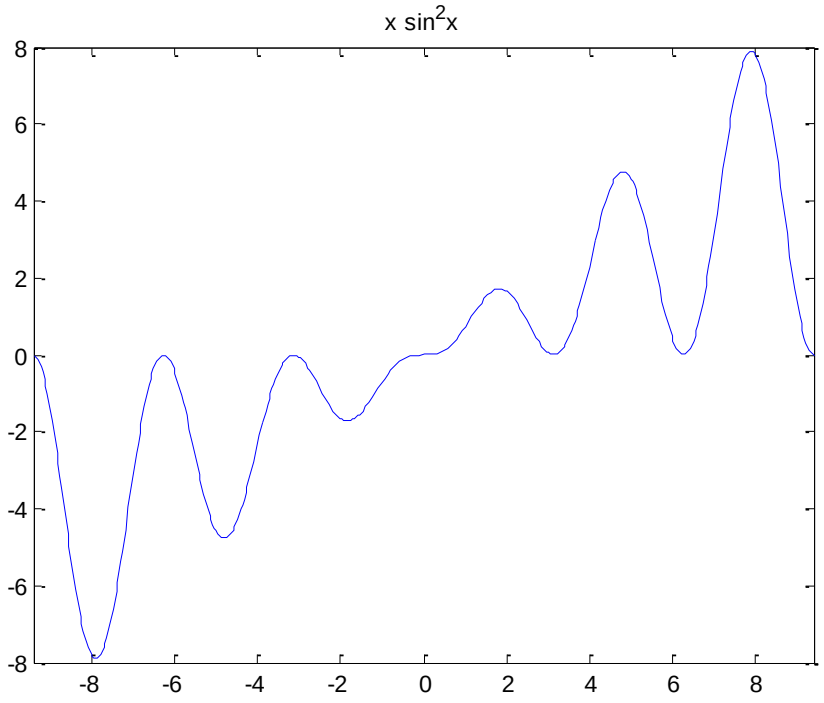


$$f(x) = \frac{x^2}{x^2 - 1}$$

fplot

$$f(x) = x\sin^2 x$$

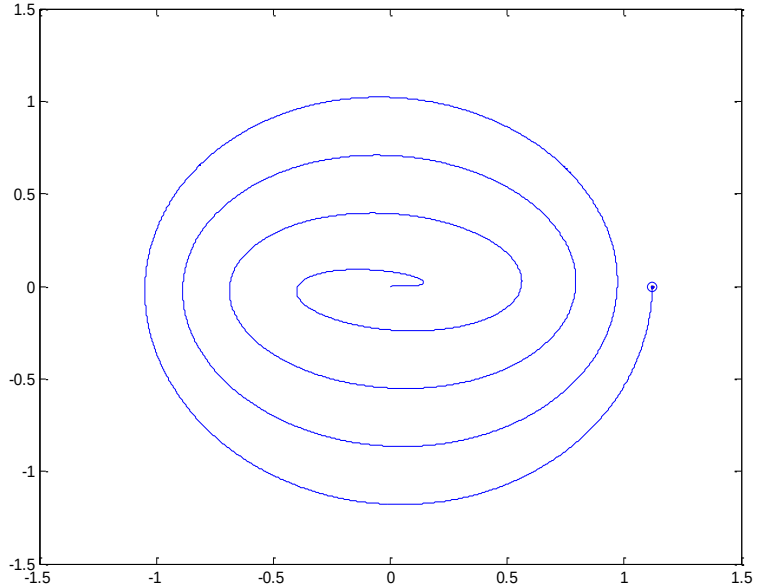
```
>>  fplot('x*sin(x)^2',[-3*pi, 3*pi]), title('x sin^2x')
```



comet

comet(x,y)

```
>> t=0:pi/1000:8*pi;
>> x=sqrt(t/20).*cos(t); y=t.*sin(t)/20;
>> comet(x,y)
```



$$x(t) = \sqrt{\frac{t}{20}}\cos t$$

$$y(t) = \frac{t\sin t}{20}$$

```
>> plot(x,y), hold on, comet(x,y)
```

Colors, lines, and symbols

plot(x,y, ' [color][stype][ltype]').

[color]	Color	Color
b	blue	**blue**
g	green	**green**
r	red	**red**
c	cyan	**cyan**
m	magenta	**magenda**
y	yellow	**yellow**
k	black	**black**
w	white	

[ltype]	Line type
-	solid
:	dotted
--	dashed
-.	dashdot

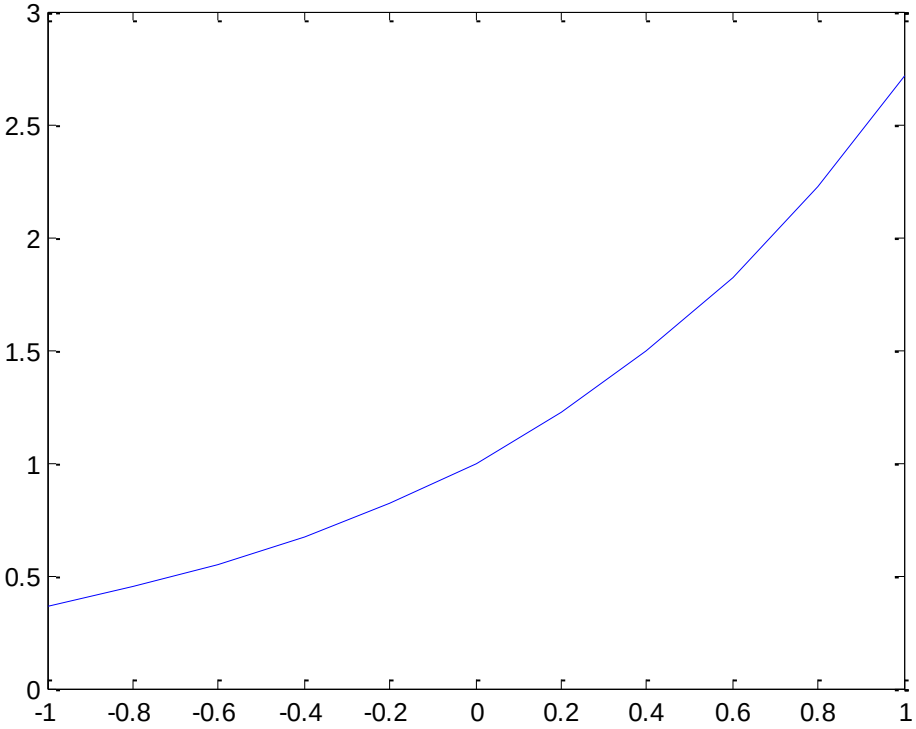
[stype]	Symbol
.	point
o	circle
x	x-mark
+	plus
*	star
s	square
d	diamond
v	triangle (down)
^	triangle (up)
<	triangle (left)
>	triangle (right)
p	pentagram
h	hexagram

Plot

```
>> x = -1:0.2:1;
>> y=exp(x);
>> plot(x, y)
```

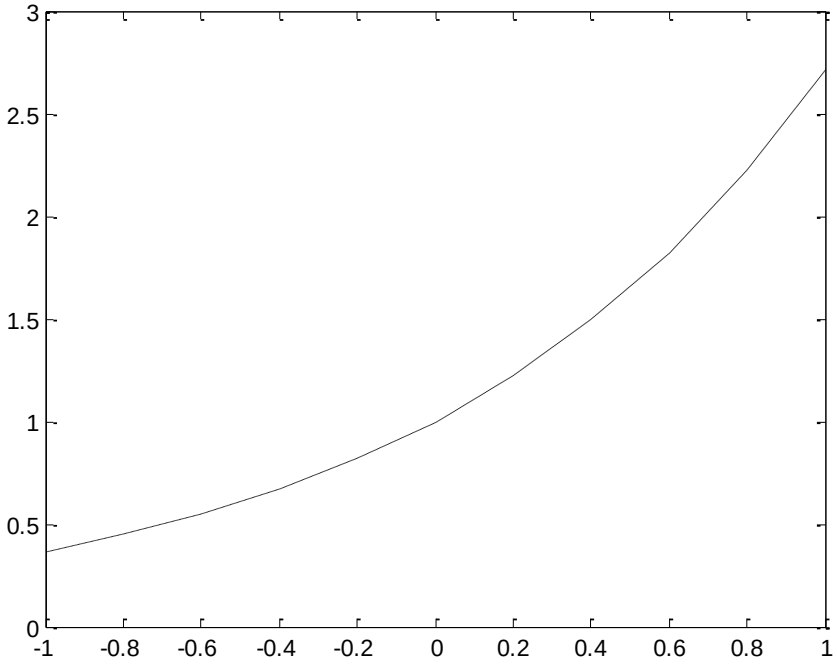
or

```
>>plot(x, y, 'b- ')
```



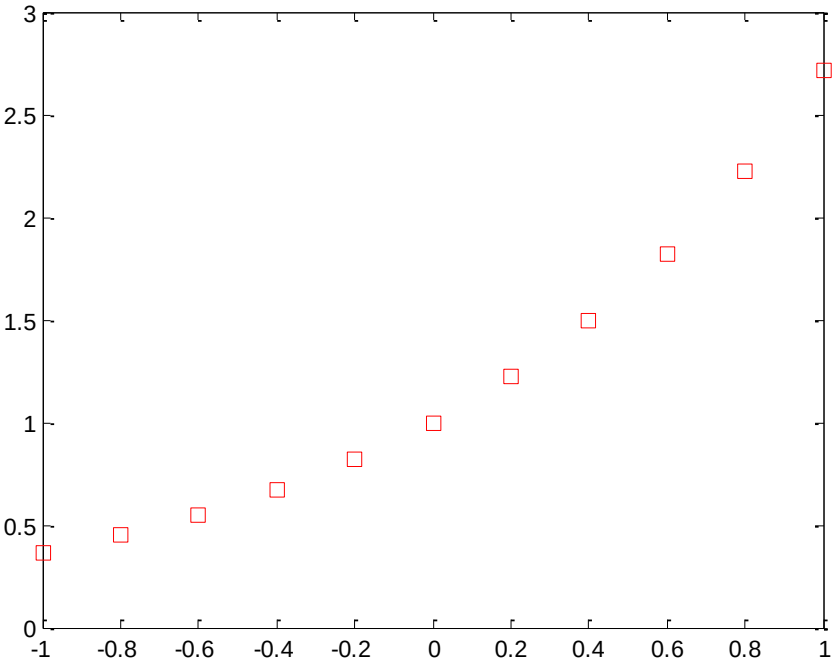
Plot

```
>> plot(x, y, 'k--')
```



Plot

```
>>  plot( x, y, 'rs')
```

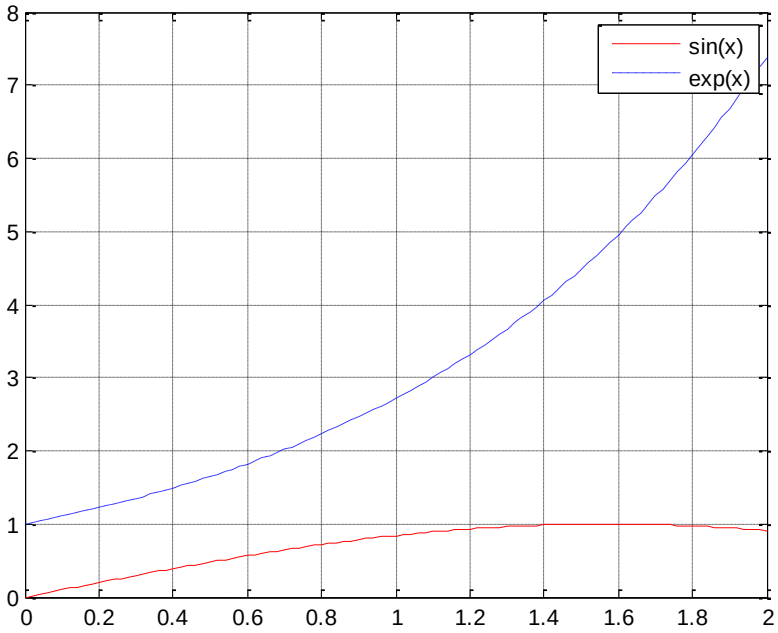


Multiple graphs

plot(x1, y1, ' [colour][stype][ltype]', x2, y2, ' [colour][stype][ltype]', …….)

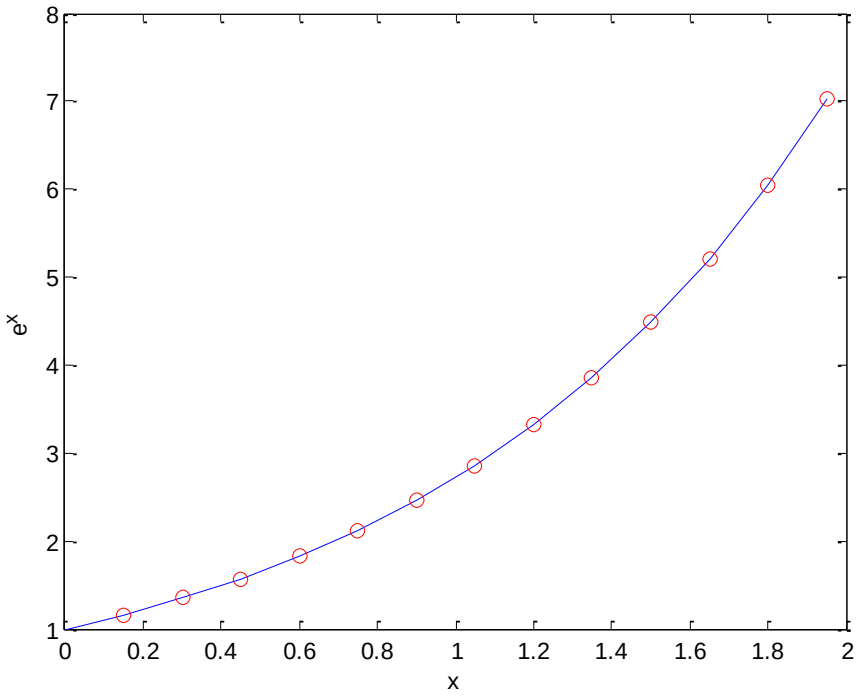
legend ('legend y1', 'legend y2', ……)

```
>> x=0:0.02:2;
>> y=sin(x);
>> z=exp(x);
>> plot( x, y, 'r', x, z, '--')
>> grid
>> legend ( 'sin(x)', 'exp(x)' )
```



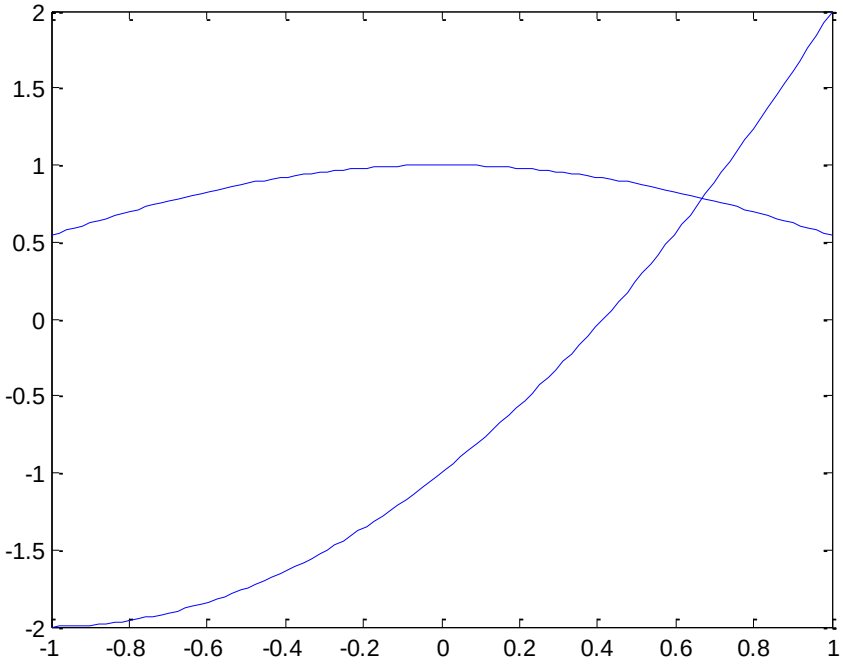
Plot

```
>> x=0:0.15:2;
>> y=exp(x);
>> plot( x, y, 'b', x, y, 'ro')
>> xlabel('x')
>> ylabel('e^x')
```



hold

```
>> t=linspace(-1,1);
>> y=t.^2 + 2*t -1;
>> plot(t,y)

>> hold on
>> z=cos(t)
>> plot(t,z)
>> hold off
```



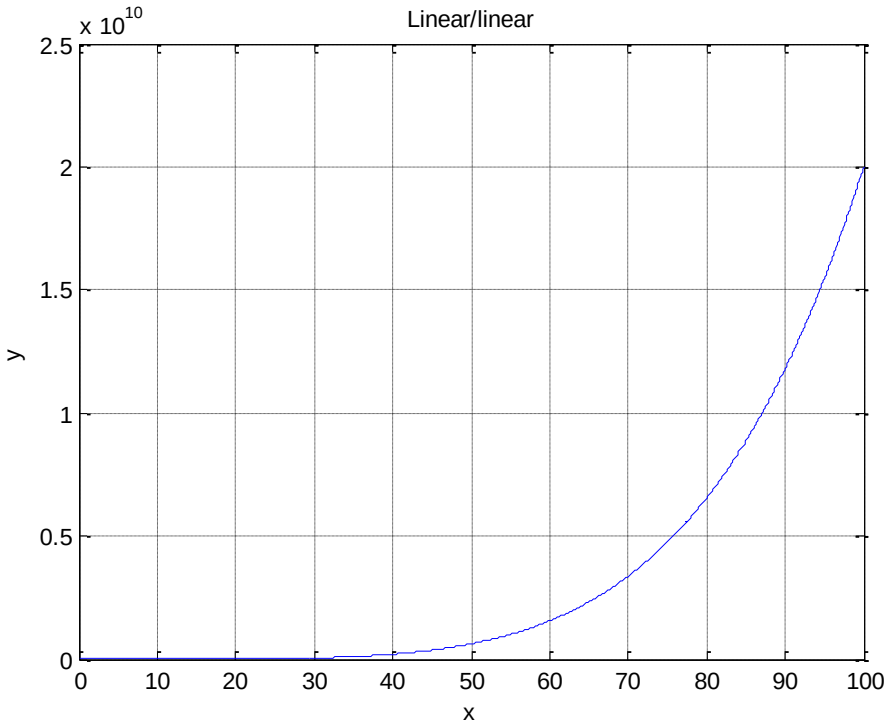
Logarithmic plots

```
plot(x,y)

loglog(x,y)

semilogx(y)

semilogy(x,y)
```

$$y = 1 + x^5$$

plot

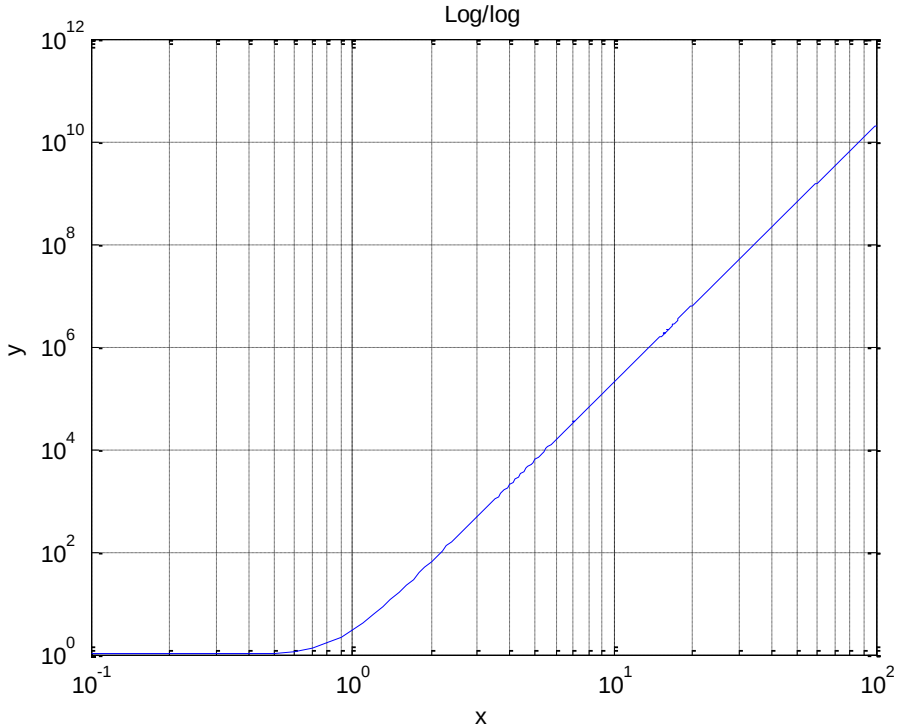
```
>> x=0:0.1:100;
>> y=1+2*x.^5;
 >> plot(x,y), grid, xlabel('x'), ylabel('y')
>> title('Linear/linear')
```



$$y = 1 + x^5$$

loglog(x,y)

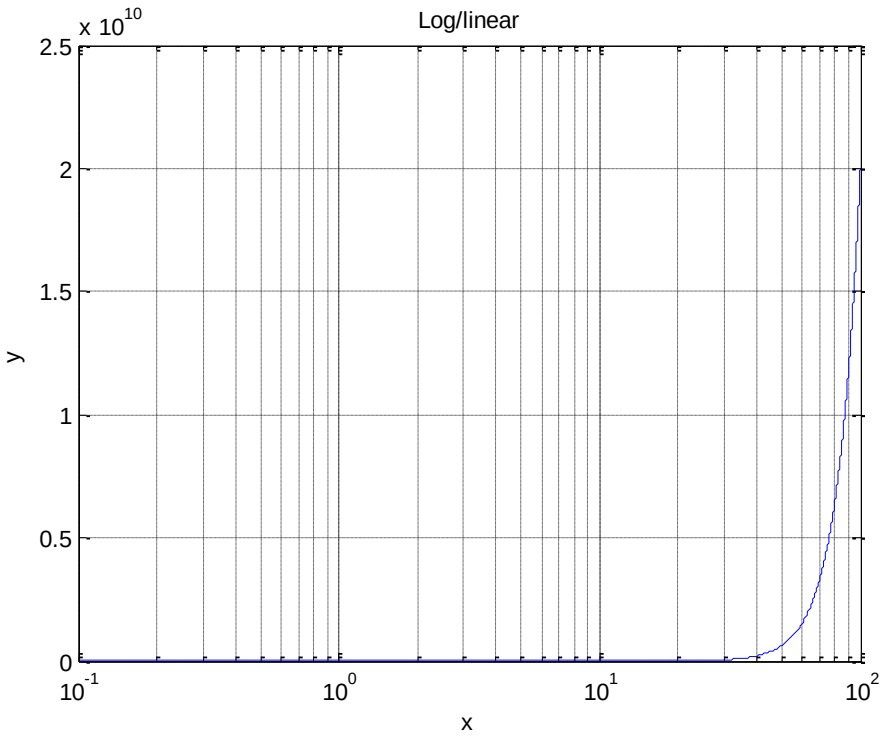
```
>> loglog(x,y), grid, xlabel('x'), ylabel('y')
>> title('Log/log')
```



$$y = 1 + x^5$$

semilogx(x,y)

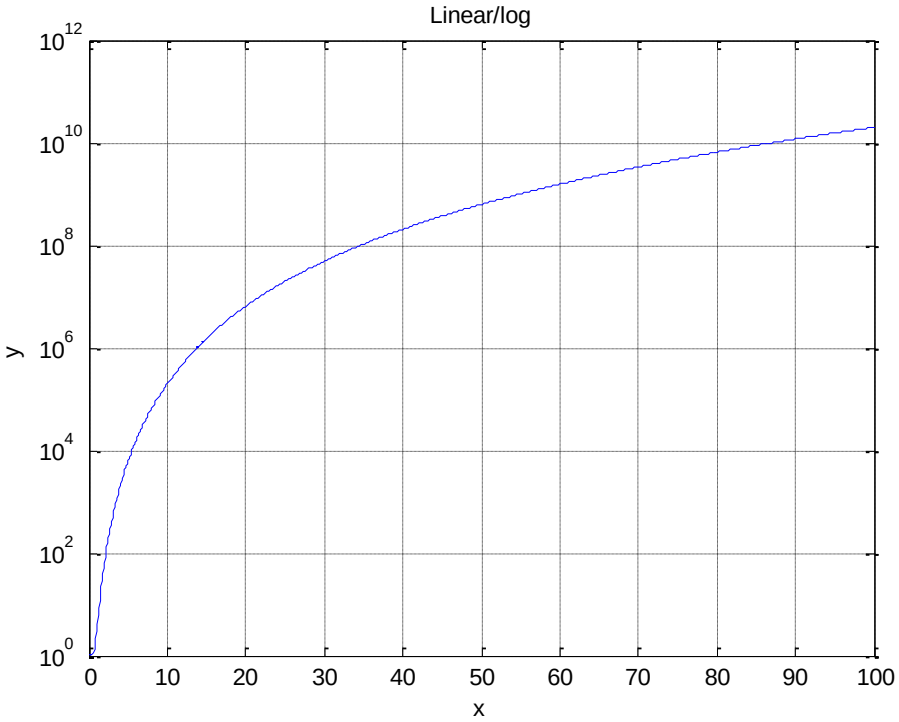
```
>> semilogx(x,y), grid, xlabel('x'), ylabel('y')
>> title('Log/linear')
```



$$y = 1 + x^5$$

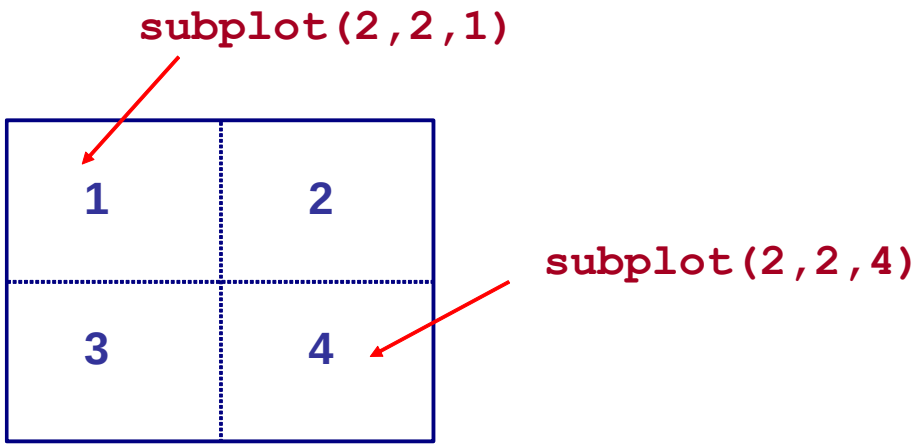
semilogy(x,y)

```
>> semilogy(x,y), grid, xlabel('x'), ylabel('y')
>> title('Linear/log')
```



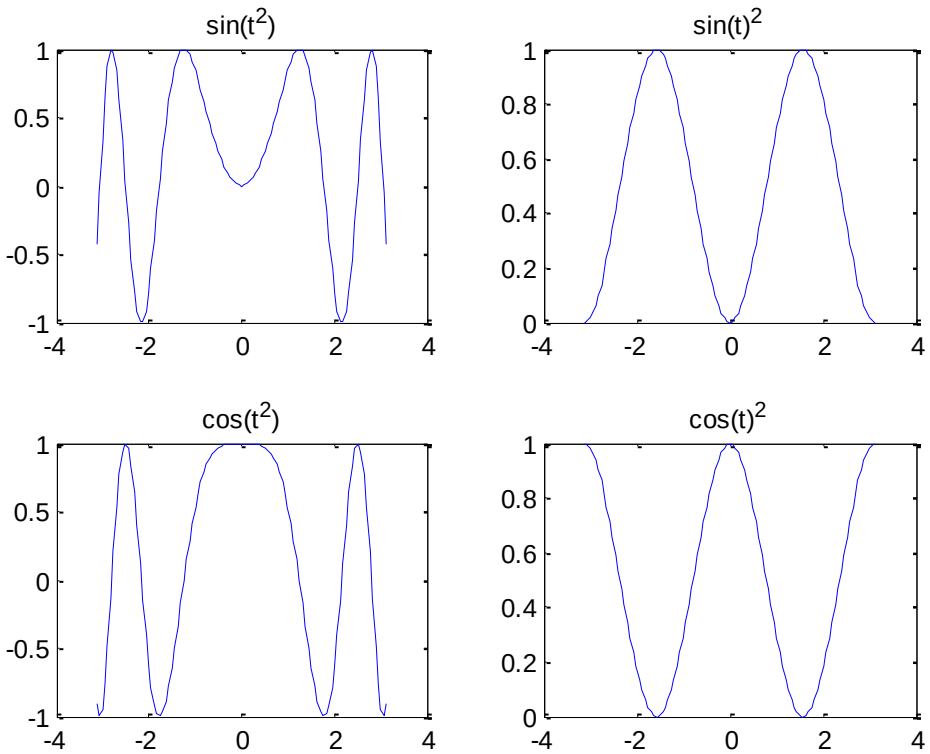
Multiple plots

subplot(m,n,p)



Example

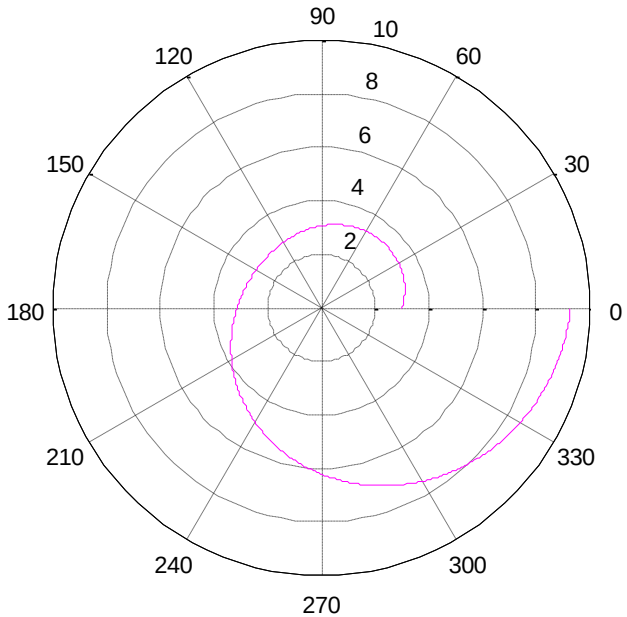
```
>> t = -pi:2*pi/100:pi;
>>f1=sin(t.^2);
>>f2=(sin(t)).^2;
>>f3=cos(t.^2);
>>f4=(cos(t)).^2;
>>subplot(2,2,1);plot(t,f1);
>>title('sin(t^2)')
>>subplot(2,2,2);plot(t,f2);
>>title('sin(t)^2')
>>subplot(2,2,3);plot(t,f3);
>>title('cos(t^2)')
>>subplot(2,2,4);plot(t,f4);
>>title('cos(t)^2')
```



Graphs in polar coordinates

polar(theta,r)

```
>> t=0:0.01:2*pi;
>> r=3*cos(t/2).^2+t;
>> polar(t,r, 'm')
```

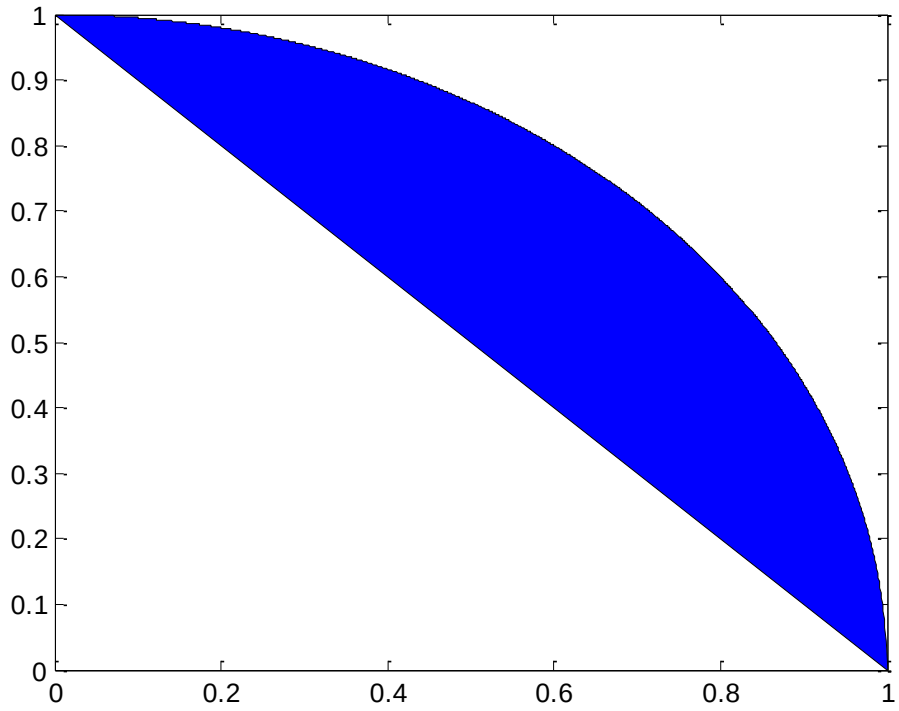


$$r = 3\cos^2\frac{\theta}{2} + \theta$$

fill

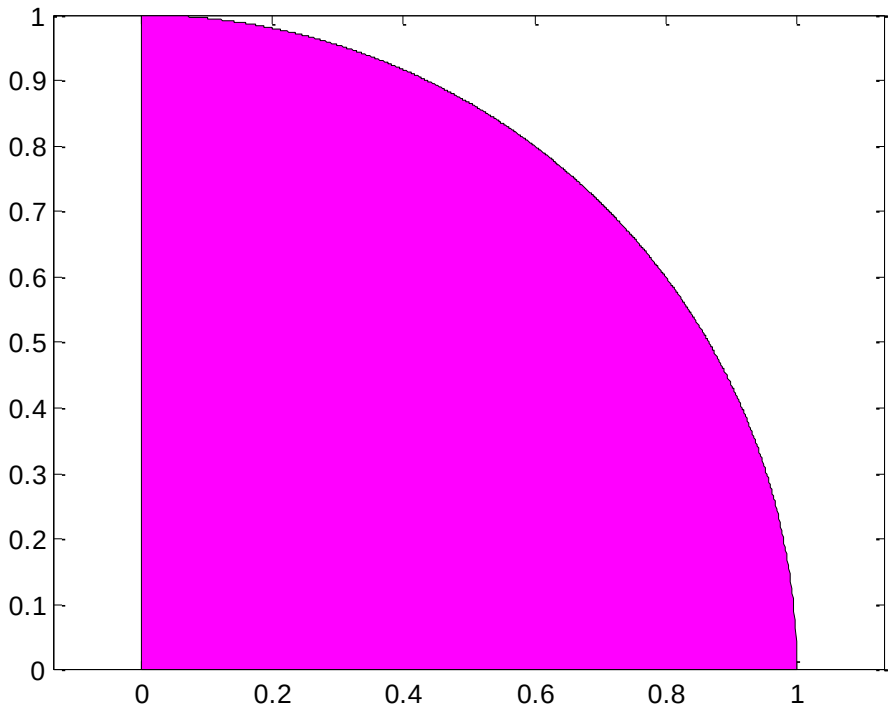
fill(X,Y,C) fills the 2-D polygon defined by vectors X and Y with the color specified by C.

```
>> x=linspace(0,1,1001); y=sqrt(1-x.^2);
>> fill(x,y,'b')
```



fill

```
>> x=linspace(0,1,1001); y=sqrt(1-x.^2);
>> X=[x 0]; Y=[y 0];
>> fill(X,Y,'m'), axis equal
```

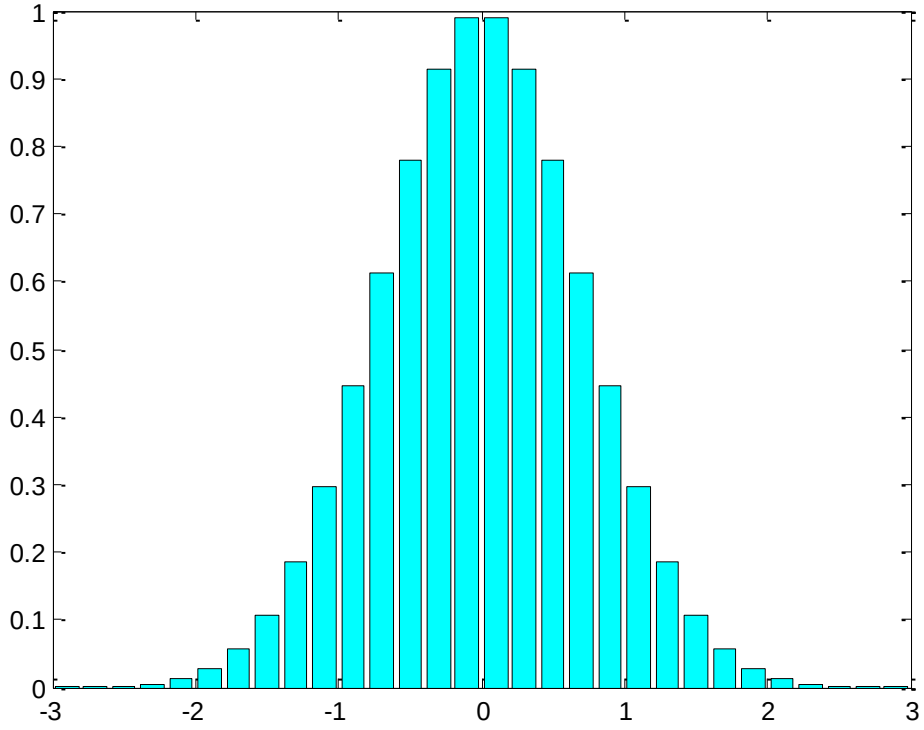


Bar and area graphs

```
bar(x)

barh(x)

bar3(x)

bar3h(x)

area(x)
```

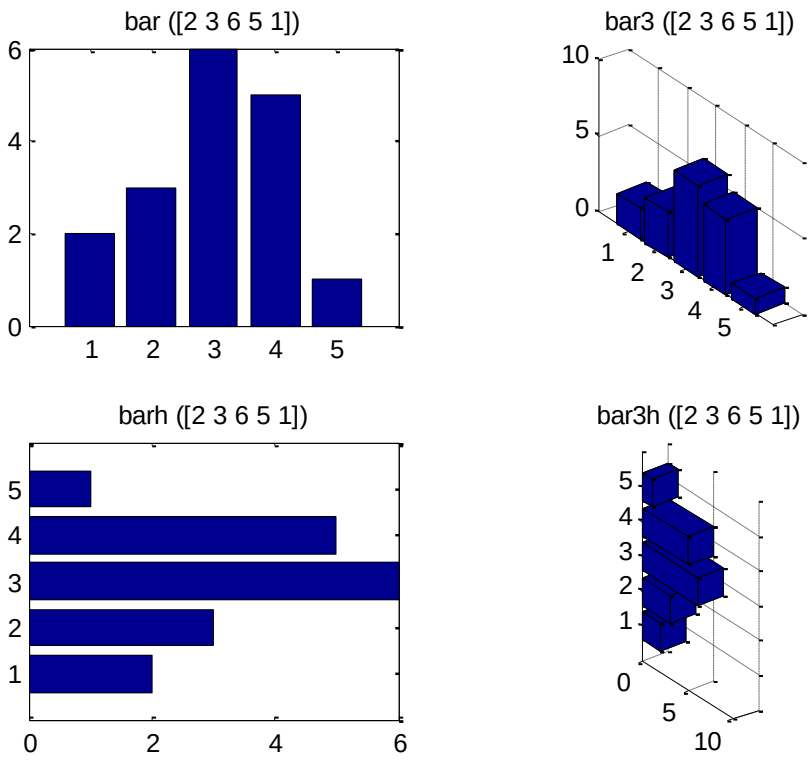
Example 1

```
>> x = -2.9:0.2:2.9;
>> y=exp(-x.^2);
>> bar(x,y)
>> colormap cool
```



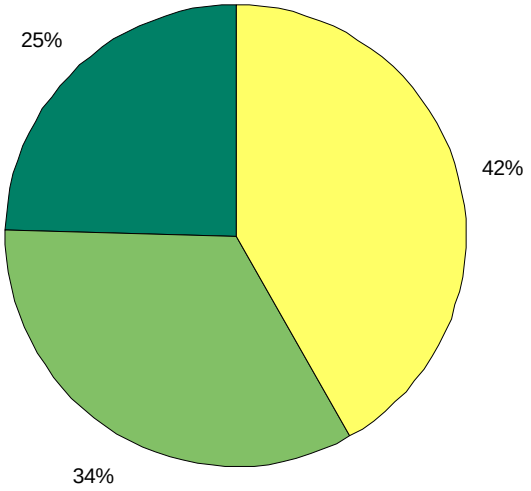
Example 2

```
>> x=[2 3 6 5 1];
>> subplot(2,2,1), bar(x), title('bar ([2 3 6 5 1])')
>> subplot(2,2,2), bar3(x), title('bar3 ([2 3 6 5 1])')
>> subplot(2,2,3), barh(x), title('barh ([2 3 6 5 1])')
>> subplot(2,2,4), bar3h(x), title('bar3h ([2 3 6 5 1])')
```

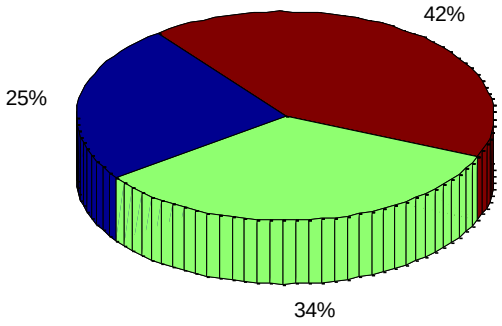


Pie charts

```
>> x=[194.8,266.5,330.9];
>> pie(x)
```

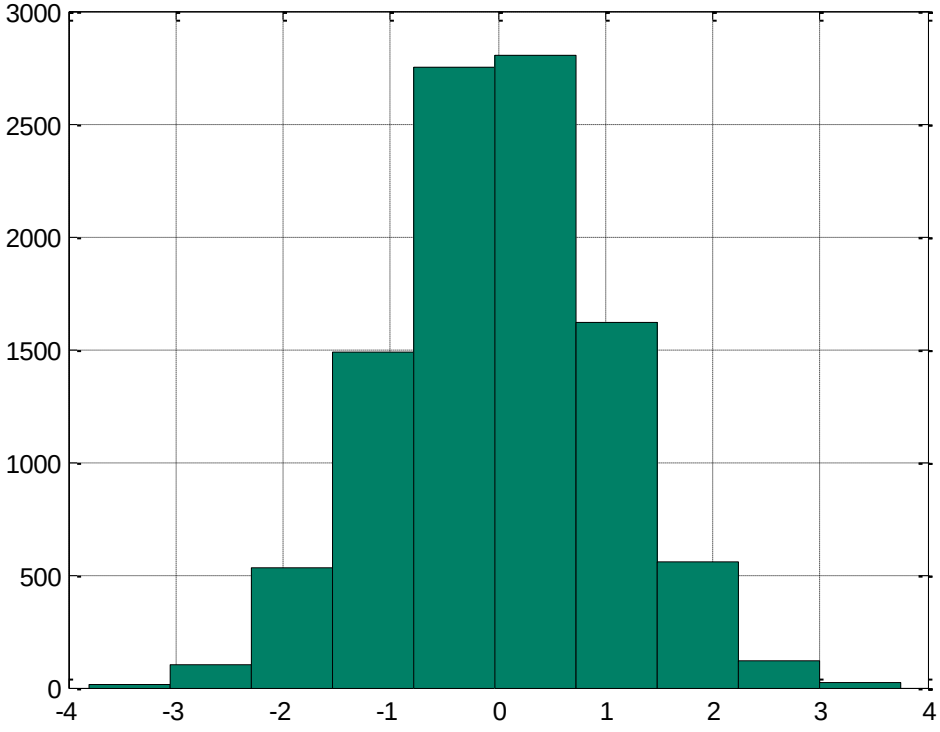



```
>> pie3(x)
```



Histograms

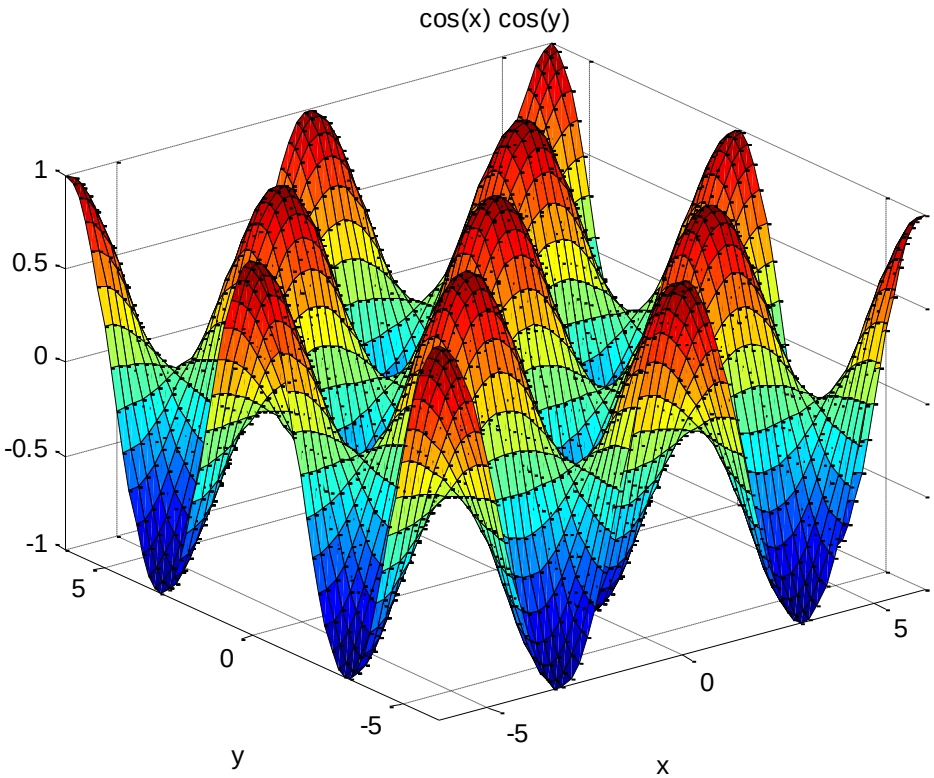
```
>> y=randn(10000,1);
>> hist(y)
>> grid
```



3D plots

ezsurf

```
>> z = @(x,y) cos(x).*cos(y);
>> ezsurf(z)
```

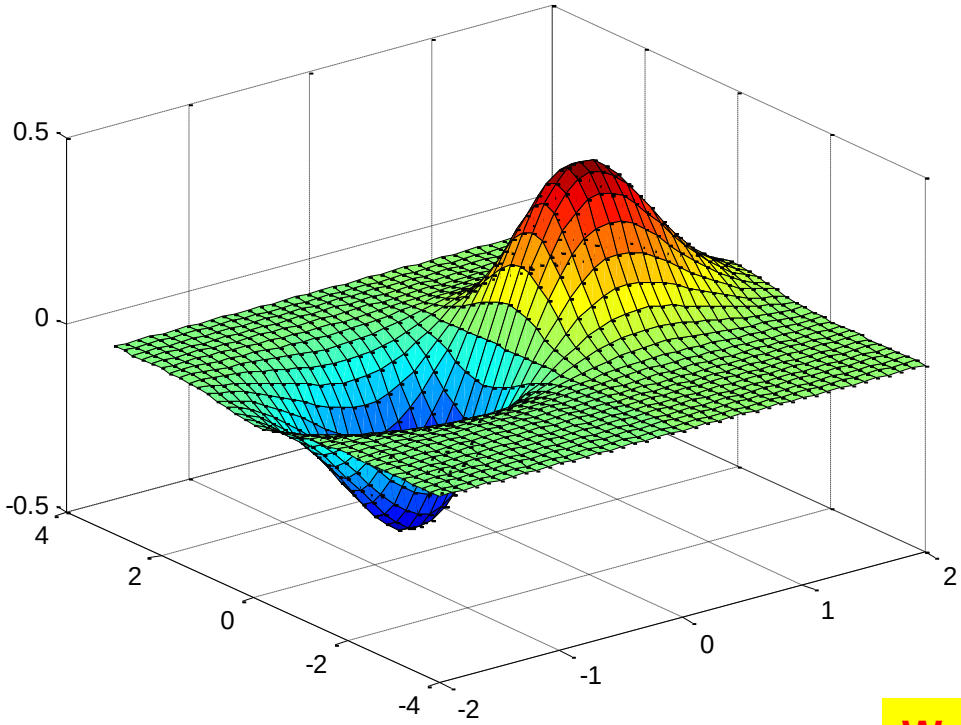


$$z = \cos x \cos y$$

3D plots

meshgrid and surf

```
>> [x,y] = meshgrid(-2:0.1:2, -4:0.2:3);
>> z = x .* exp(-x.^2 - y.^2);
>> surf(x,y,z)
```



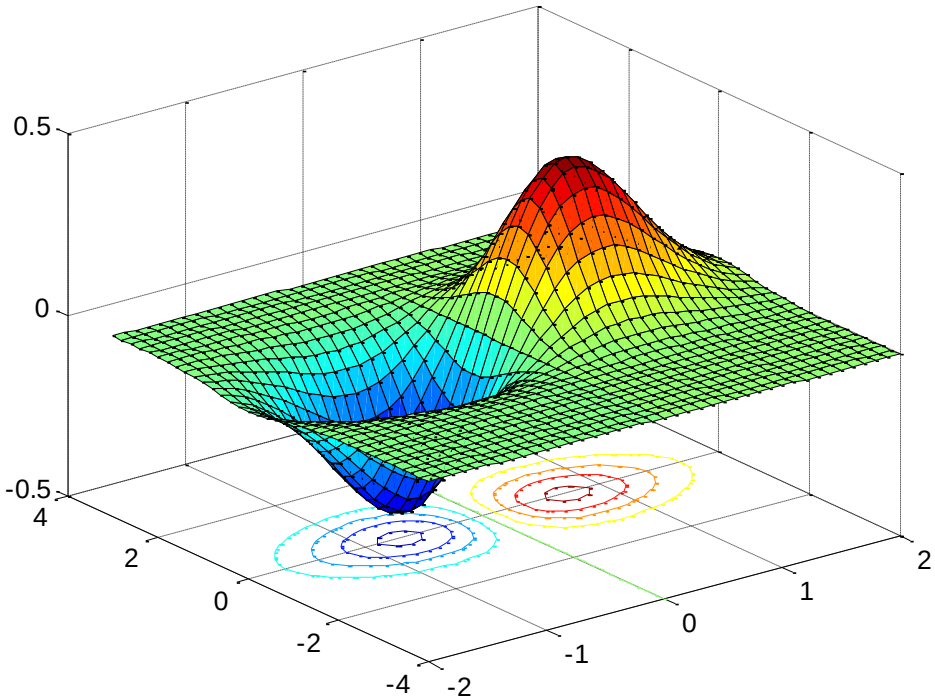
$$z = xe^{-x^2-y^2}$$

We can put axis labels!

3D plots

surfc

```
>> [x,y] = meshgrid(-2:0.1:2, -4:0.2:3);
>> z = x .* exp(-x.^2 - y.^2);
>> surfc(x,y,z)
```

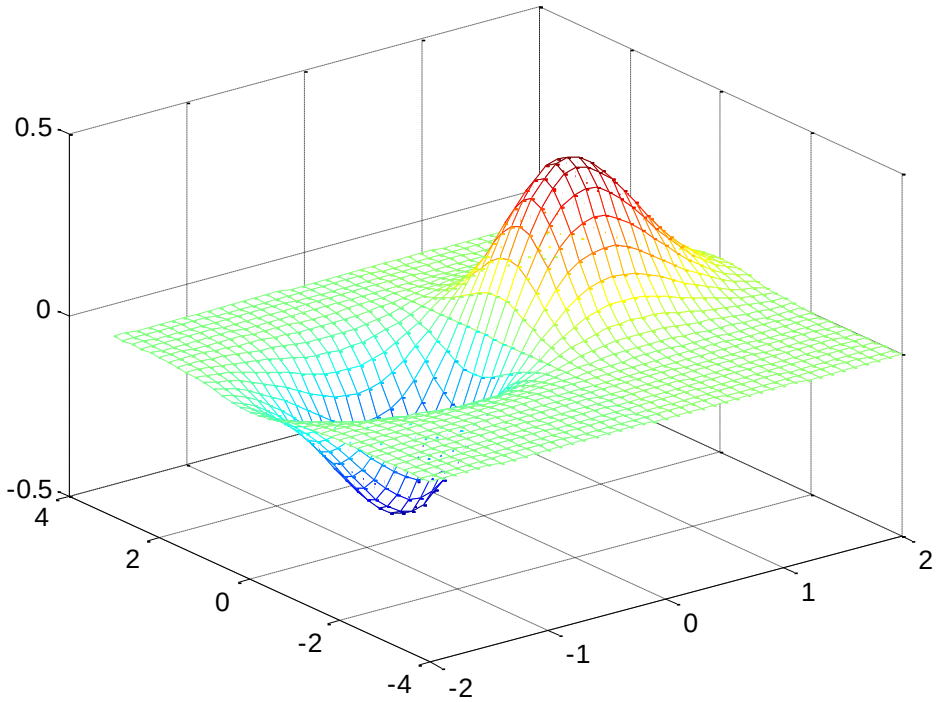


$$z = xe^{-x^2-y^2}$$

3D plots

mesh

```
>> [x,y] = meshgrid(-2:0.1:2, -4:0.2:3);
>> z = x .* exp(-x.^2 - y.^2);
>> mesh(x,y,z)
```

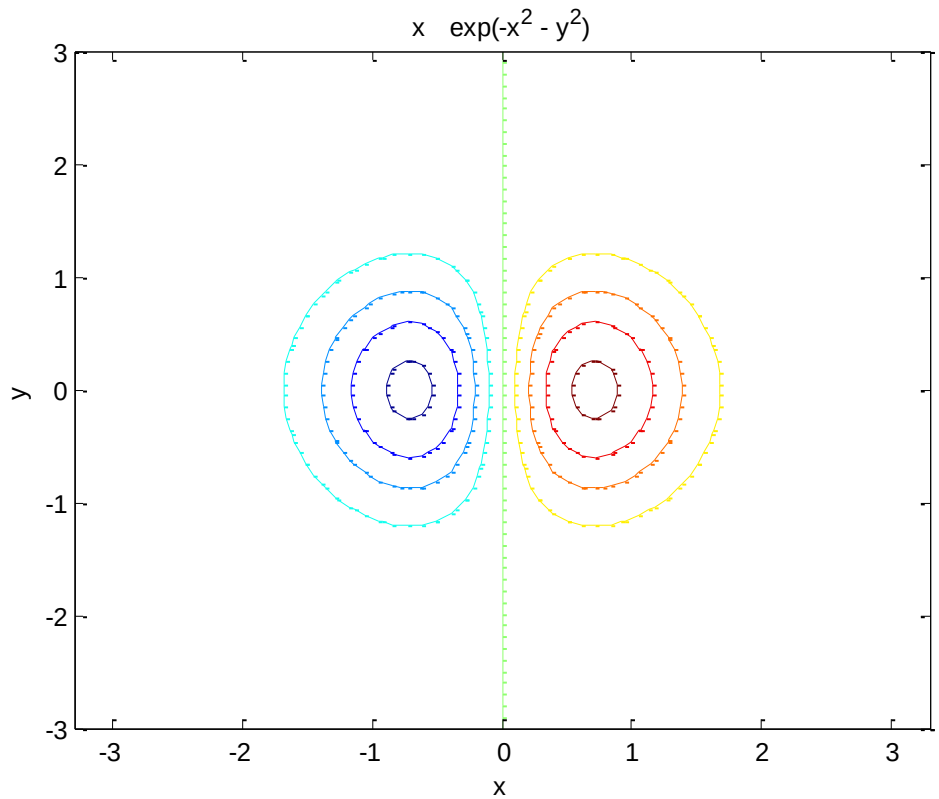


$$z = xe^{-x^2-y^2}$$

Contour plots

ezcontour

```
>>  z = @(x,y) x .* exp(-x.^2 - y.^2);
>> ezcontour(z)
```

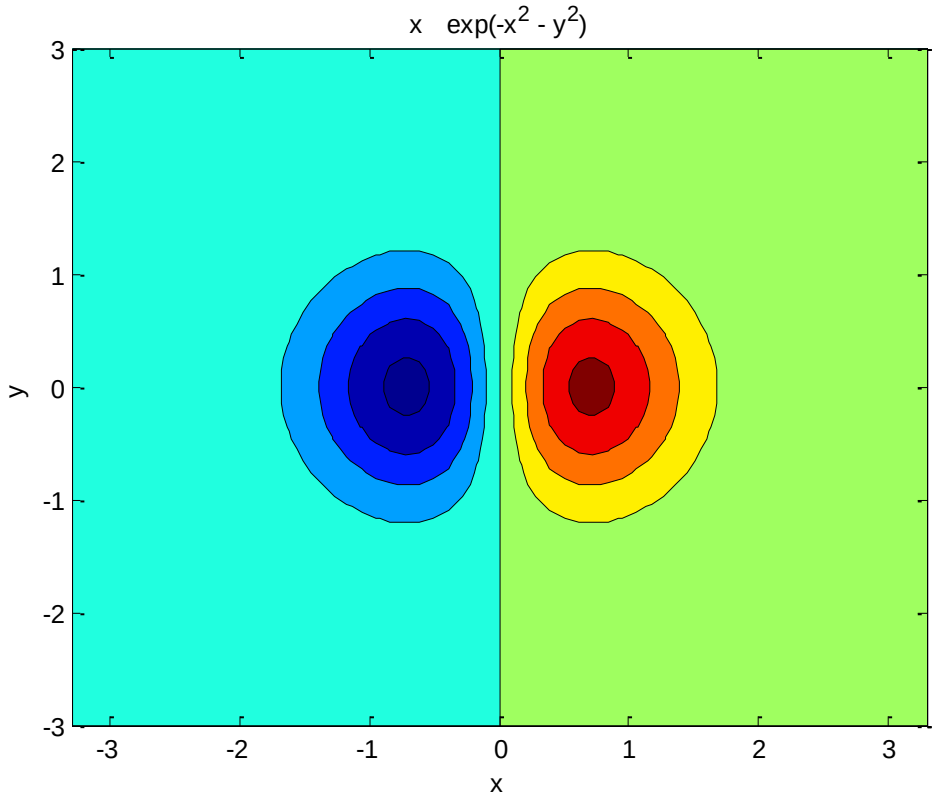


$$z = xe^{-x^2-y^2}$$

Contour plots

ezcontourf

```
>>  z = @(x,y) x .* exp(-x.^2 - y.^2);
>> ezcontourf(z)
```

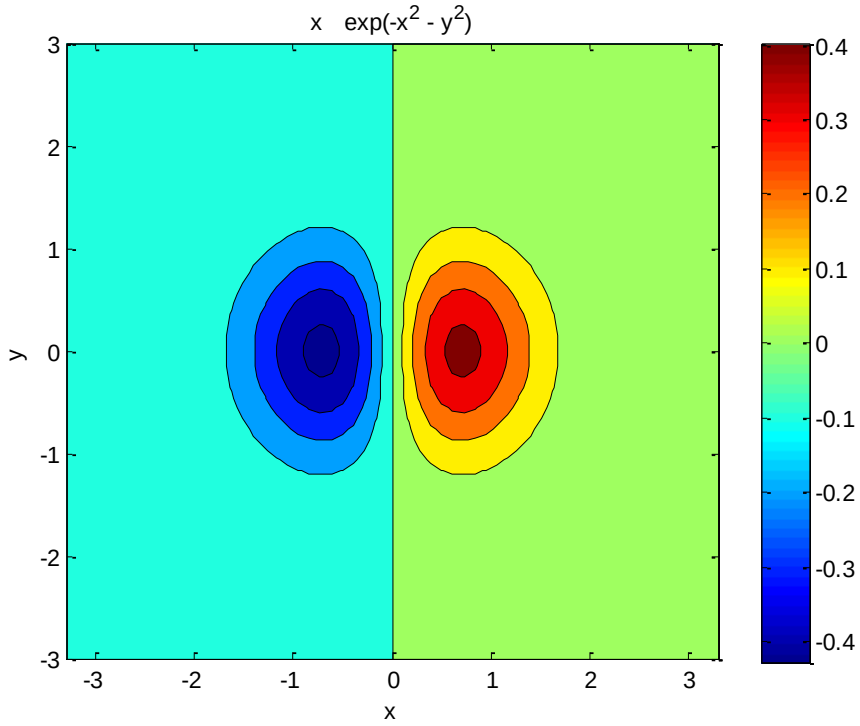


$$z = xe^{-x^2-y^2}$$

Contour plots

colorbar

```
>>  z = @(x,y) x .* exp(-x.^2 - y.^2);
>> ezcontourf(z)
>> colorbar
```

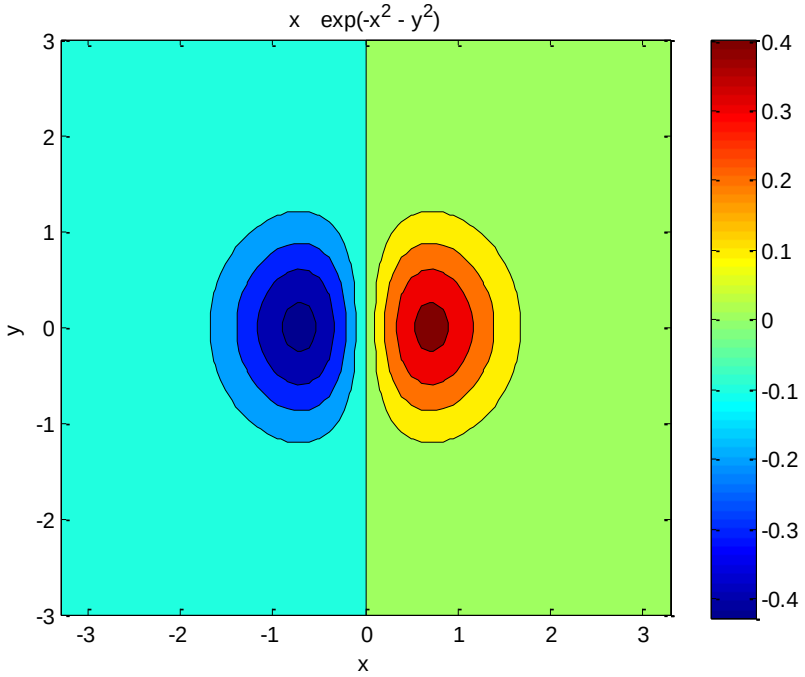


$$z = xe^{-x^2-y^2}$$

Η εντολή

```
>> colorbar
```

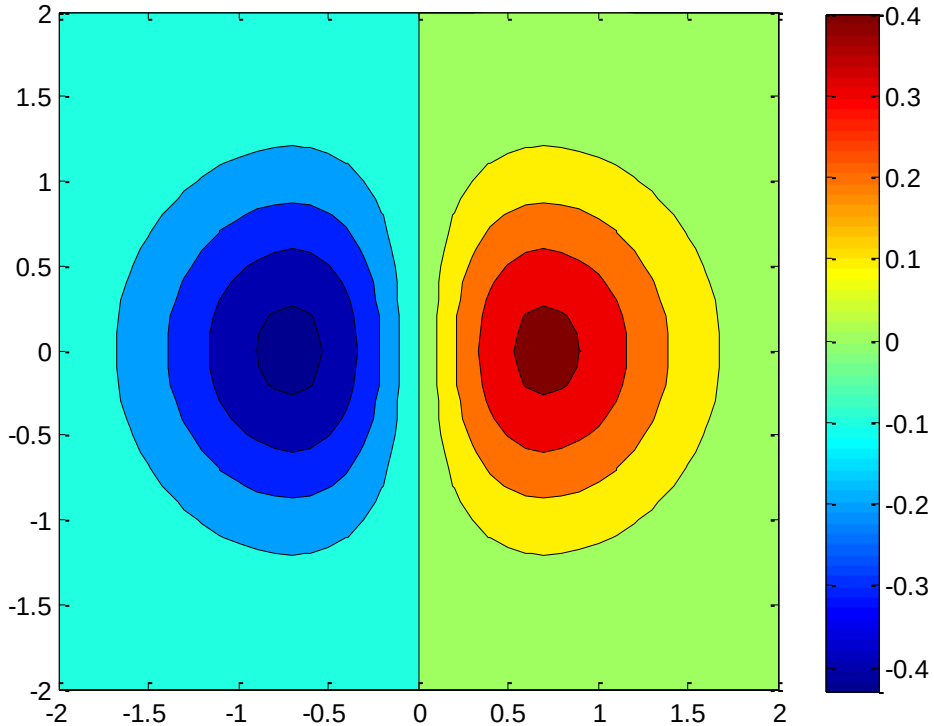
προσθέτει λεζάντα για το κάθε χρώμα, όπως φαίνεται πιο κάτω:



Contour plots

contour, contourf

```
>>  [x,y] = meshgrid(-2:0.1:2, -2:0.1:2);
>> z = x .* exp(-x.^2 - y.^2);
>> contourf(x,y,z), colorbar
```



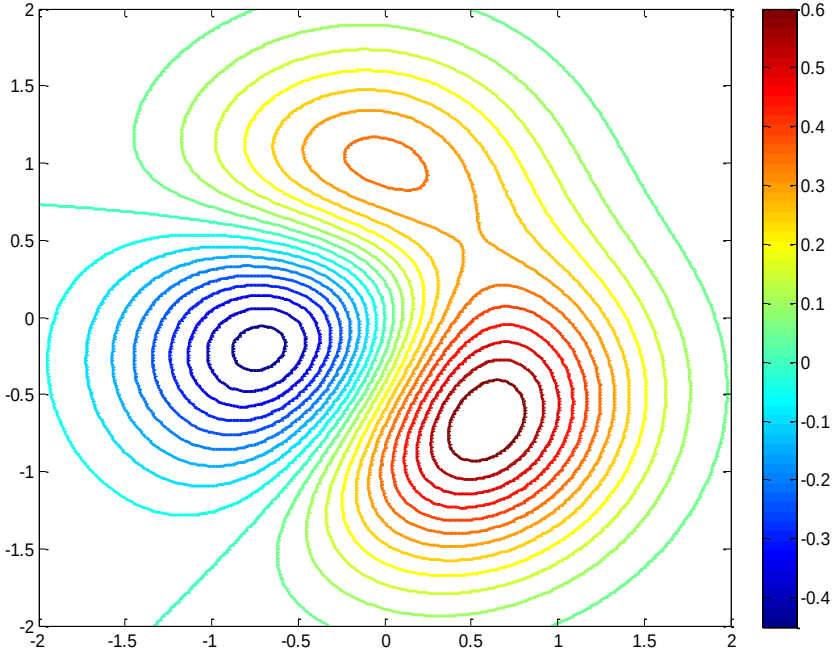
$$z = xe^{-x^2-y^2}$$

contour, contourf

```
contour(x,y,z,n)

contour(x,y,z,v)

contourf(x,y,z,v)

contour(x,y,z,[v v])

contourf(x,y,z,[v v])
```

Example 1

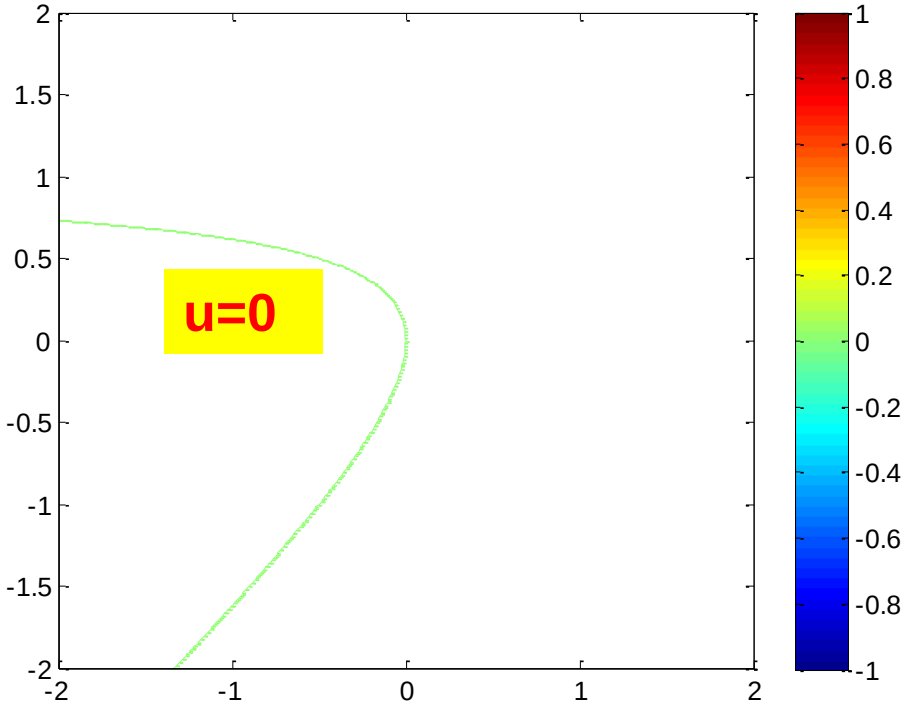
```
>> [x,y]=meshgrid(-2:0.02:2,-2:0.02:2);
>> z=(x-x.*y+y.^2).*exp(-x.^2-y.^2);
>> contour(x,y,z,-.5:0.05:5,'Linewidth',2), colorbar
```



$$z = (x - xy - y^2)e^{-x^2-y^2}$$

Example 2

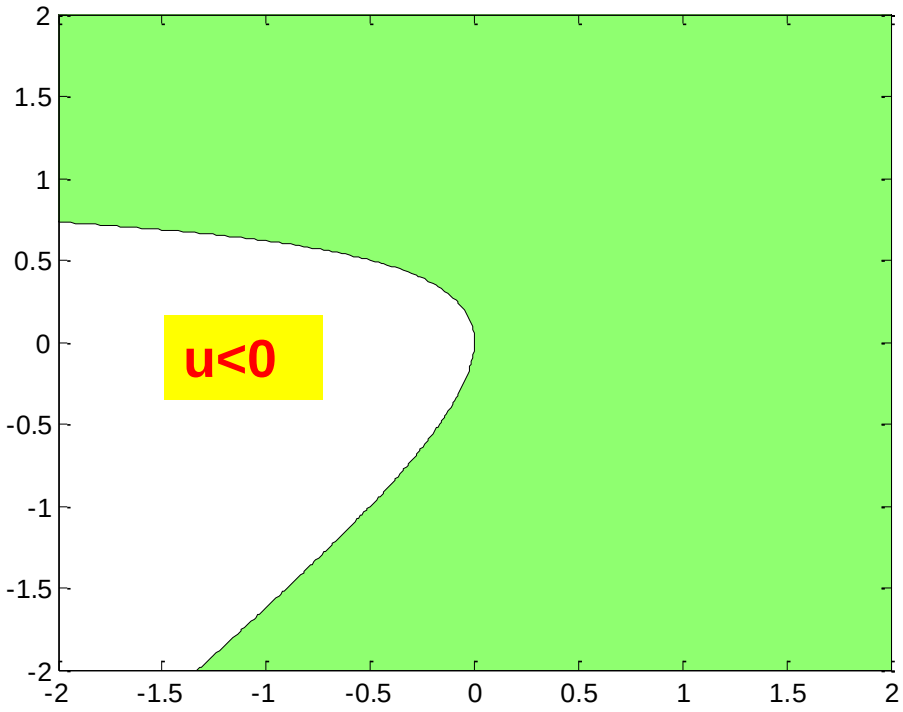
```
>> [x,y]=meshgrid(-2:0.02:2,-2:0.02:2);
>> z=(x-x.*y+y.^2).*exp(-x.^2-y.^2);
>> contour(x,y,z,[0 0]), colorbar
```



$$z = (x - xy - y^2)e^{-x^2-y^2}$$

Example 3

```
>> [x,y]=meshgrid(-2:0.02:2,-2:0.02:2);
>> z=(x-x.*y+y.^2).*exp(-x.^2-y.^2);
>> contourf(x,y,z,[0 0])
```

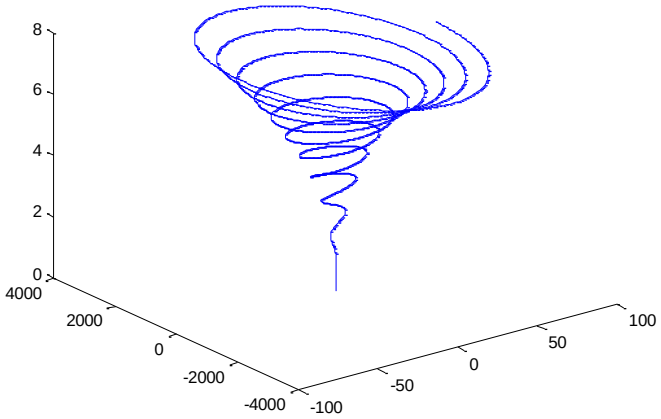


$$z = (x - xy - y^2)e^{-x^2-y^2}$$

3D curves: plot3

$$x(t) = t\cos t, \quad y(t) = t^2 \sin t, \quad z(t) = \sqrt{t}, \quad t \in [0, 20\pi]$$

```
t=0:pi/100:20*pi;
x=t.*cos(t);
y=t.^2.*sin(t);
z=sqrt(t);
plot3(x,y,z)
```



comet3

```
>> t=0:pi/100:20*pi;
>> x=t.*cos(t);
>> y=t.^2.*sin(t);
>> z=sqrt(t);
>> comet3(x,y,z)
```

Storing a figure in a file: print

print -device -options filename

Examples

print -djpeg -r150 figu
Store current figure into ‘figu.jpg’ with a 150 digit resolution

print
Sends the current figure to your current printer.

print –dps ‘foo’
Save the current figure to a postscript file named 'foo.ps'

Device options

-dwinc : Send figure to current printer in color
-dmeta : Send figure to clipboard (or file) in Metafile format
-dpsc : PostScript for color printers
-dpsc2 : Level 2 PostScript for color printers
-depsc : Encapsulated Color PostScript
-depsc2 : Encapsulated Level 2 Color PostScript
-djpeg<nn> : JPEG image, quality level of nn
-dtiff : TIFF with packbits (lossless run-length encoding) compression

print -depsc -tiff -r300 matilda

Saves current figure at 300 dpi in color EPS to matilda.eps with a TIFF preview

See help print for more info!

Thank you!!