PROGRAMMING WITH MATLAB

WEEK 6

PLOTTING

PLOTTING

- Plot:

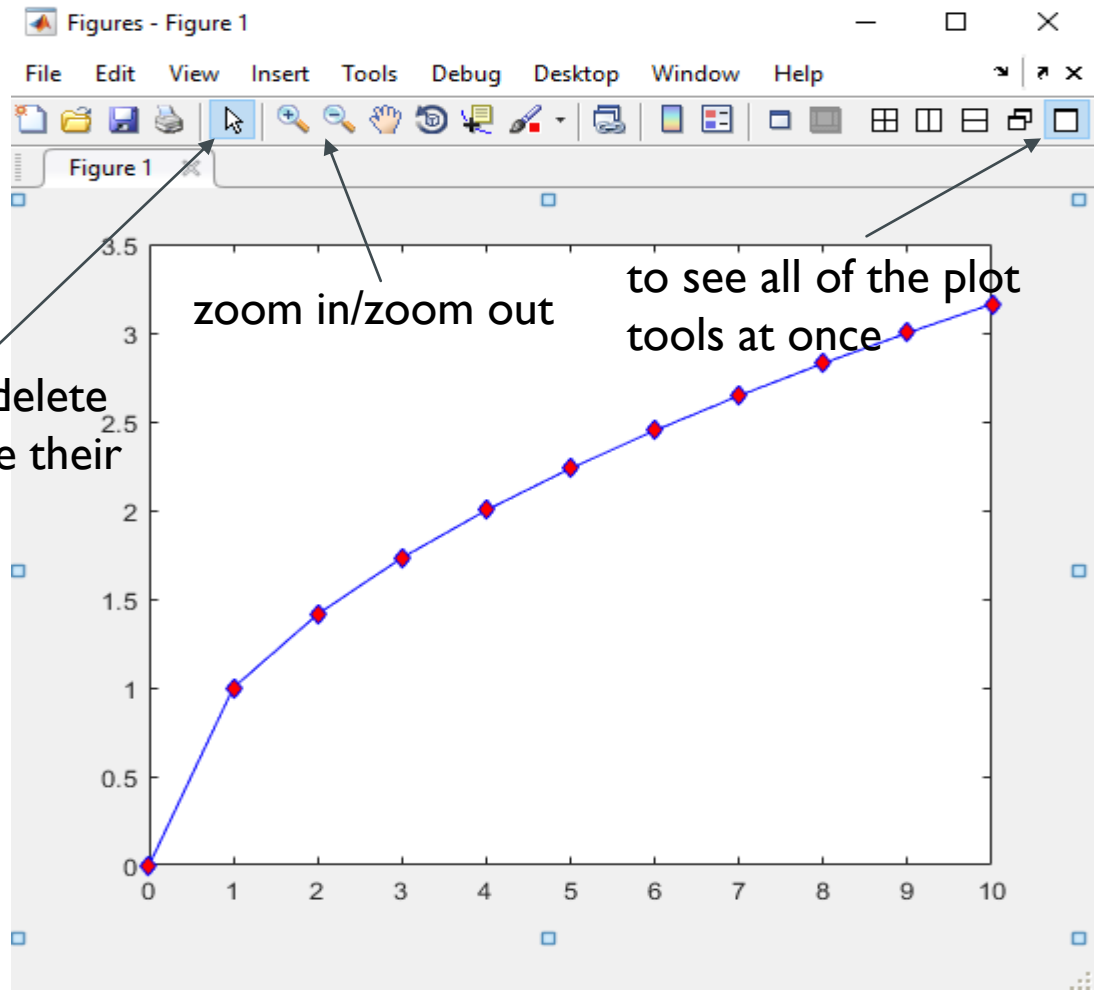
Syntax: plot(x, y, 'r.-')

Color

Marker

Linestyle

- The line color, marker style and line style can be changed by adding a string argument.



PLOTTING

Line Style	Description
-	Solid line
--	Dashed line
:	Dotted line
-.	Dash-dot line

Color	Description
y	Yellow
m	Magenta
c	Cyan
r	Red
g	Green
b	Blue
w	White
k	black

Marker	Description
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross

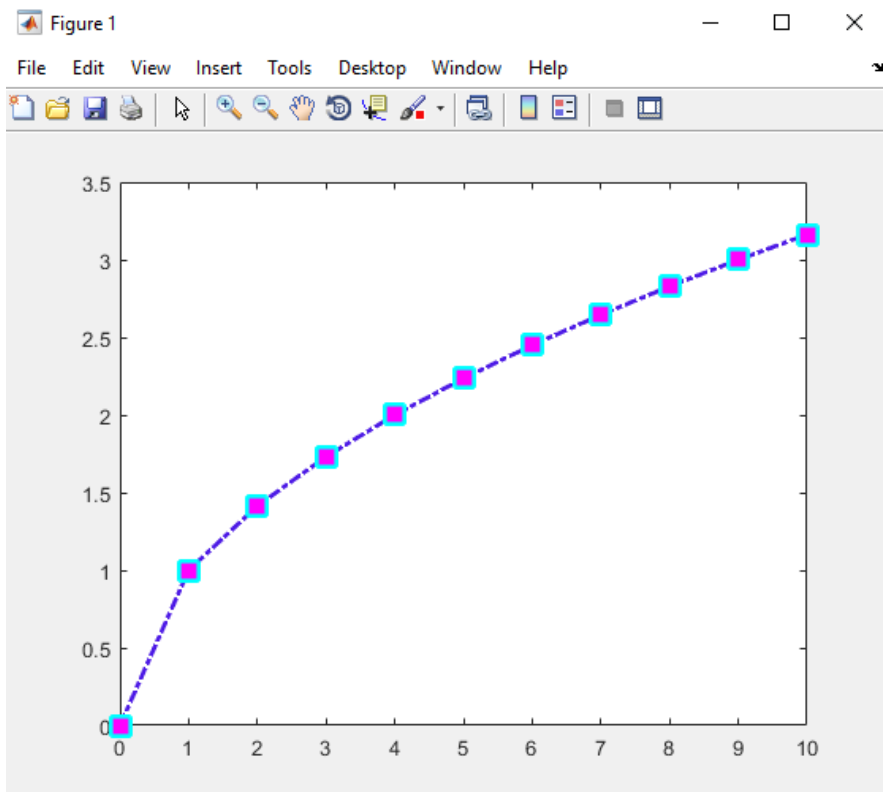
Marker	Description
S	Square
D	Diamond
^	Upward pointing triangle
v	Downward pointing triangle
>	Right pointing triangle

Marker	Description
<	Left pointing triangle
P	Pentagram
h	Hexagram

PLOTTING

- All features related to a line can be customized.

```
>>  plot(x, y, '-.s', 'LineWidth', 2, ...
            'Color', [.3 .1 .9], ...
            'MarkerEdgeColor', 'c', ...
            'MarkerFaceColor', 'm', ...
            'MarkerSize', 12)
```



You can set colors by specifying [R G B] values or by using default color characters such as 'r', 'g'.

PLOTTING

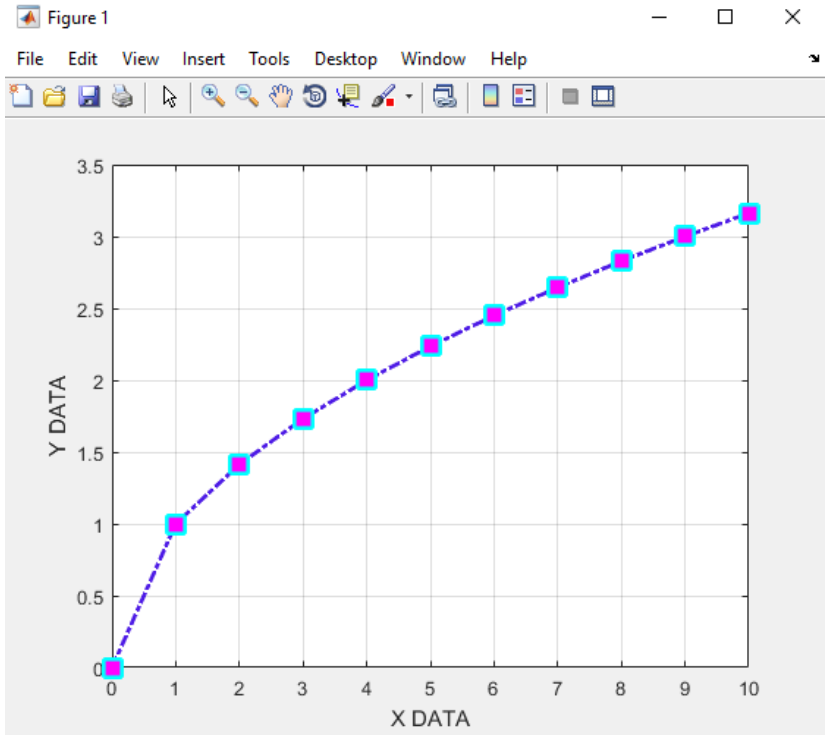
Each axis should be labeled to describe the quantity being drawn and to indicate the unit.

If you plot more than one data set, label each one or use a legend so that they can be distinguished.

The "grid" command displays gridlines at the tick marks corresponding to the tick labels.

Type "grid on" to insert gridlines; type "grid off" to stop drawing grid lines

```
>> plot(x, y, '-.s', 'LineWidth', 2, ...
                'Color', [.3 .1 .9], ...
                'MarkerEdgeColor', 'c', ...
                'MarkerFaceColor', 'm', ...
                'MarkerSize', 12)
>> xlabel('X DATA')
>> ylabel('Y DATA')
>> grid on
```

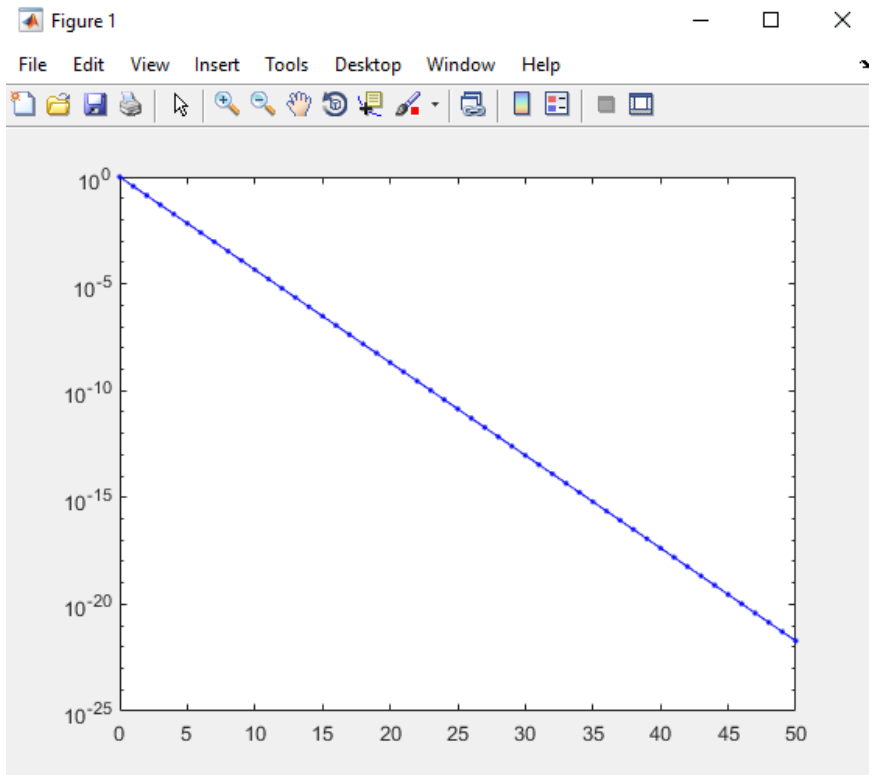


PLOTTING

- semilogx, semilogy, loglog:

Syntax: same as plot command

```
>> semilogx(x,y,’b’);
>> semilogy(y,’r.-’);
>> loglog(x,y);
>> semilogy(x, exp(-x),'b.-')
```



PLOTTING

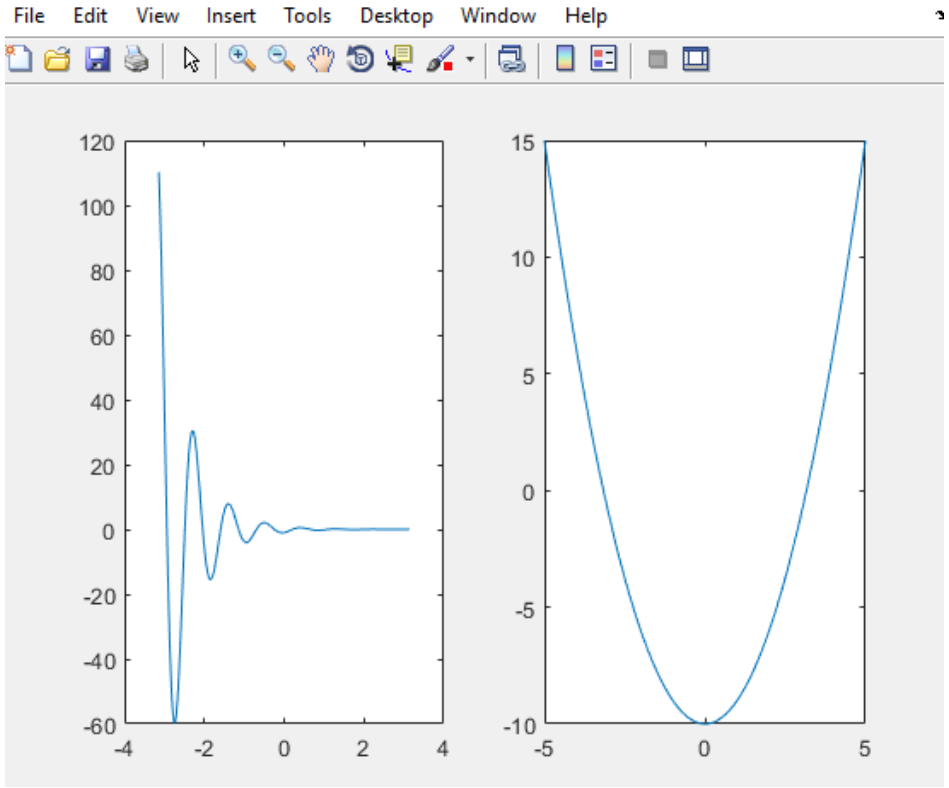
- subplots: The subplot command is used to plot smaller subplots in the same figure.

Syntax: subplot(m,n,p)

This command divides the figure window into rectangular sections with m rows and n columns

The p variable specifies which rectangular area to plot.

```
>> x = -pi:pi/100:pi;
y = exp(-1.5*x).*cos(7*x-3);
subplot(1,2,1)
plot(x,y)
x = -5:0.01:5;
y = x.^2-10;
subplot(1,2,2)
plot(x,y)
```



PLOTTING

- Some specialized plot commands:

bar(x, y) : Creates a bar chart of y versus x

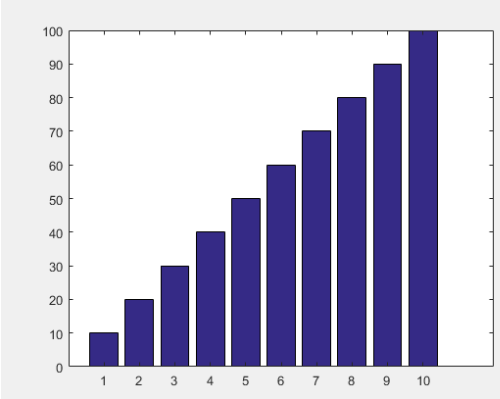
polarplot(theta,rho) : plots a line in polar coordinates, with theta indicating the angle in radians and rho indicating the radius value for each point.

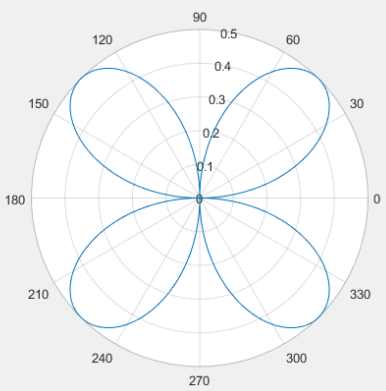
stairs(y) : draws a stairstep graph of the elements in y.

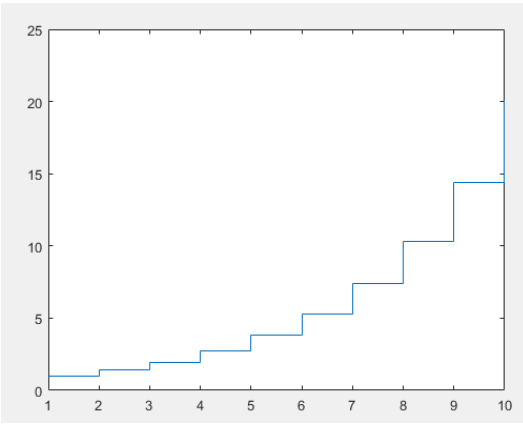
stairs(x,y) plots the elements in y at the locations specified by x.

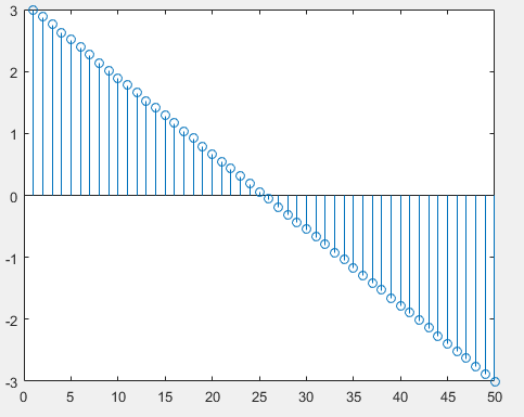
stem(y) : plots the data sequence, y as stems that extend from a baseline along the x-axis.

stem(x,y) plots the data sequence, y, at values specified by x





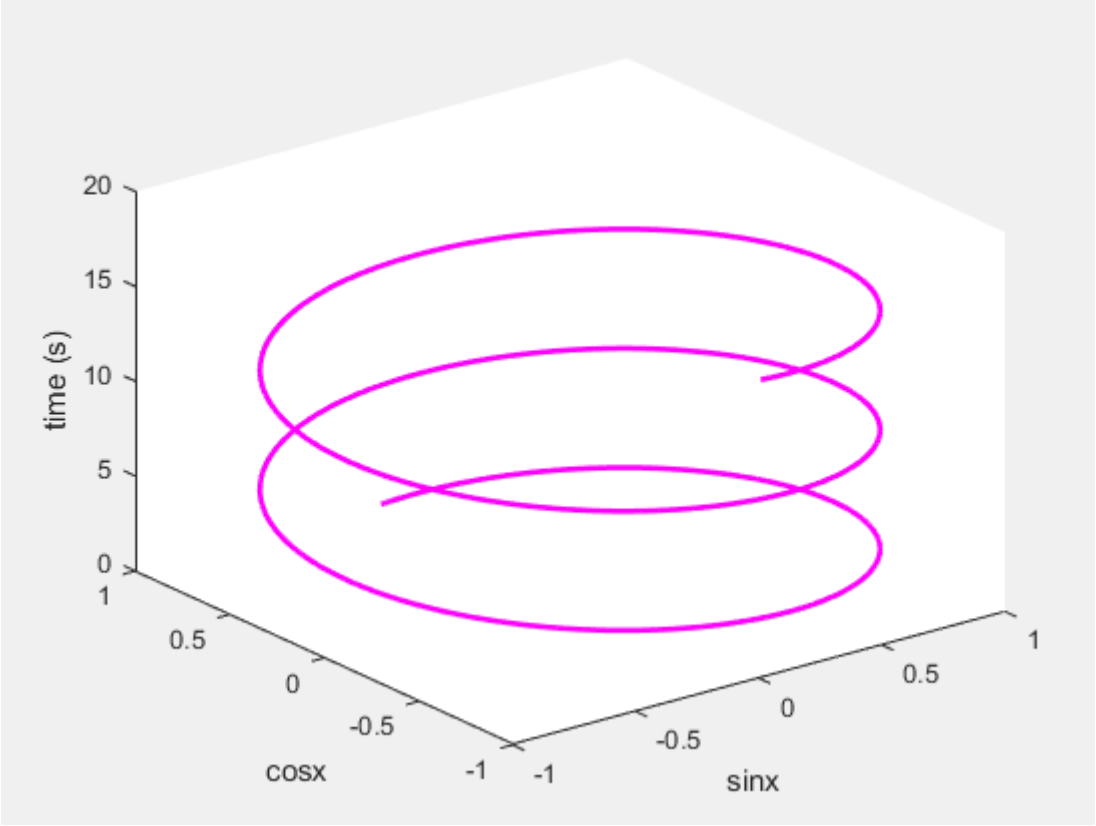




PLOTTING

Three-dimensional graphics: plot3

```
>> t = 0:0.01:5*pi;
>> x = sin(t);
>> y = cos(t);
>> z = t;
>> plot3(x,y,z,'m','LineWidth',2);
>> xlabel('sinx')
>> ylabel('cosx')
>> zlabel('time (s)')
```



PLOTTING

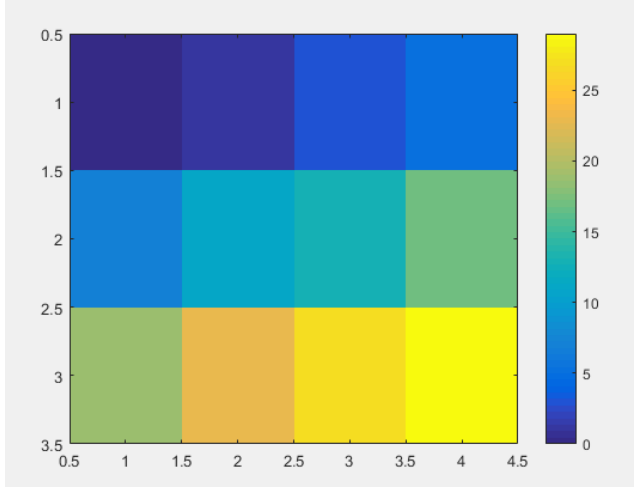
Any matrix can be visualized as an image.

imagesc(C) displays the data in array C as an image that

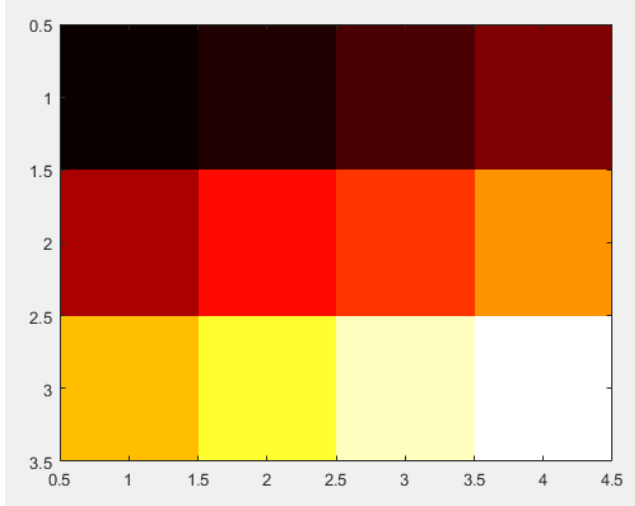
uses the full range of colors in the colormap,

automatically scales values to spread across the entire color map

```
>> C = [0 1 3 5; 7 11 13 17; 19 23 27 29];
>> imagesc(C)
>> colorbar
```




```
>> imagesc(C)
>> colormap(hot)
```



PLOTTING

Surface plots

surf places the values at the points specified in the x, y, z space,

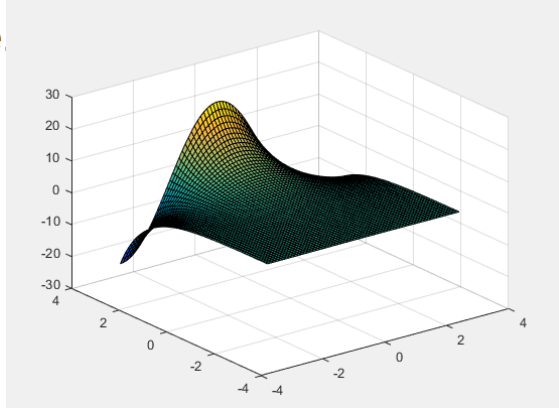
and binds all these values to obtain a surface

Values can be expressed by X, Y, Z matrices

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

>> Z = cos(X).*exp(Y)

>> surf(X,Y,Z)



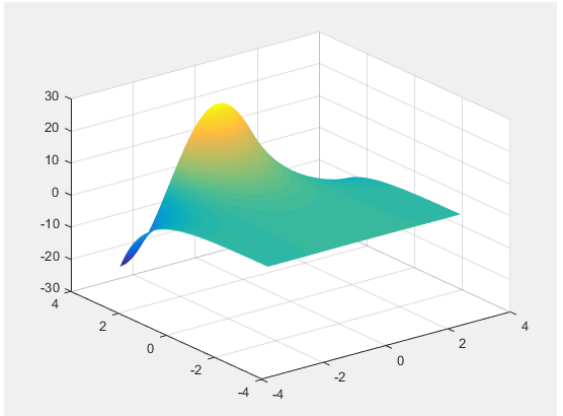
There are three types of surface color shading:

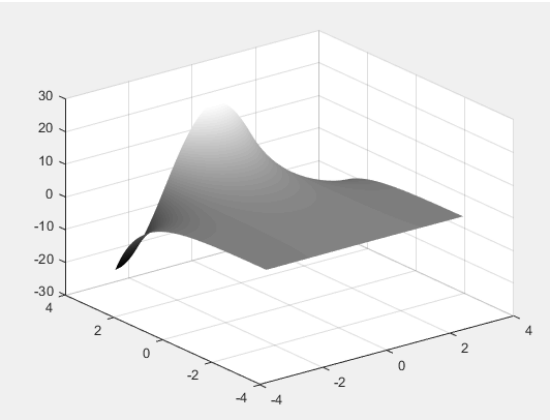
shading flat
shading faceted
shading interp

>> shading interp

You can change color maps

>> colormap(gray)





PLOTTING

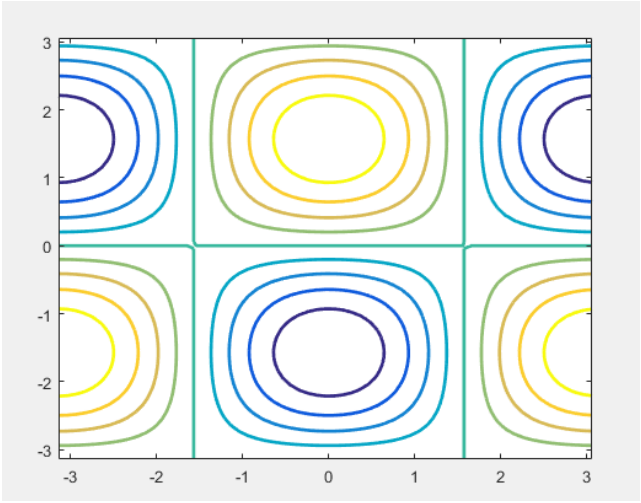
countor(X, Y, Z) : Creates a contour plot.

imagesc(C) displays the data in array C as an image that

uses the full range of colors in the colormap,

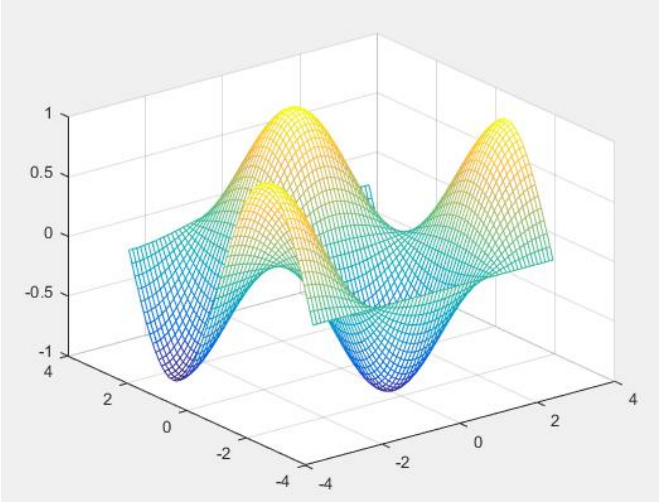
automatically scales values to spread across the entire colormap

```
>> Z = cos(X).*sin(Y);
>> contour(X,Y,Z,'LineWidth',2)
```



mesh(X,Y,Z) : Creates a 3D mesh surface plot

```
>> mesh(X,Y,Z)
```



PLOTTING

3D plotting functions

meshc(X,Y,Z) : draws a wireframe mesh and a contour plot under it with color determined by Z, so color is proportional to surface height

meshz(X,Y,Z) : draws a curtain around the wireframe mesh with color determined by Z, so color is proportional to surface height

surfc(X,Y,Z) : creates a contour plot under the three-dimensional shaded surface

waterfall(X,Y,Z) : creates a waterfall plot using the values specified in X, Y, and Z