Basics of MATLAB-II

Dr.K.Srinivasa Rao
Professor & Head
Department of Mathematics
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya
Kanchipuram, Tamilnadu

Plotting

MATLAB not only helps us for calculation but also helps us in data analysis and visualization by plotting graphs and waveforms. It provides a variety of functions for displaying data as 2-D or 3-D graphics. The commands to produce simple plots are surprisingly simple. For complicated graphs and special effects there are a lot of built-in functions that enable the user to manipulate the graphics window in many ways.

For 2-D graphics, the basic command is:

plot(xdata, ydata, 'option')

This command produces plot with xdata on the horizontal axis and ydata on the vertical axis.The 'option' is an optional argument that specifies the color, the line style (e.g.,solid,dashed,dotted), and the point-marker style (e.g., o,+,*). All three style options can be specified together. The two vectors xdata and ydata must have the same length.

Options

The 'option' in the plot command is chatacter string that consists of one, toe or three characters that specify the color and /or line style. There are several color line and marker options:

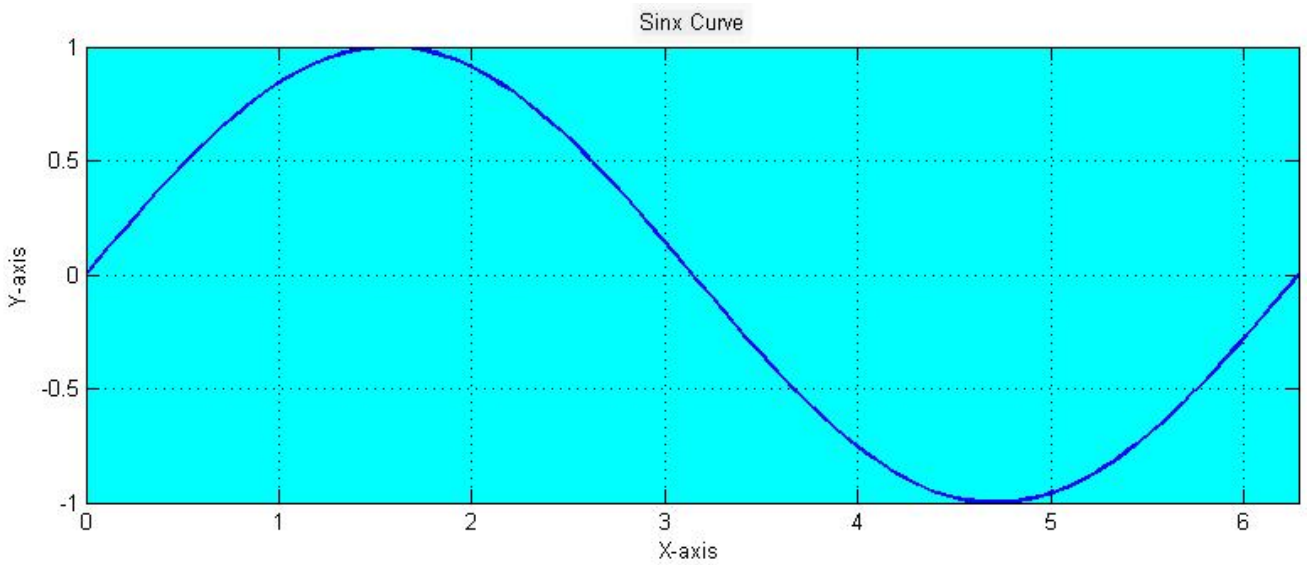
Color Option	Line Option	Marker Option
y yellow	- solid	+ plus sign
m magenta	-- dashed	o circle
c cyan	: dotted	* asterisk
r red	-. dash-dot	x x-mark
g green	none no line	. point
b blue		s square
w white		d diamand etc.,

The 'option' is made up of the ccolor option, the line option, the marker option, or combination of them. For example:

Command	**Output**
plot(x,y,'y')	plots y versus x with a yellow solid line
plot(x,y.':')	plots y versus x with a dotted line
plot(x,y,'b–')	plots y versus x with a blue dashed line
plot(x,y,'+')	plots y versus x as unconnected points marked by +

Plot $y = sinx$ in $[0, 2\pi]$

```
>> x=linspace(0,2*pi)
>> y=sin(x)
>> plot(x,y)
```



Tools Available For Manipulating a Plot

One of the primary advantages of the use of MATLAB is the range of manipulation tools available for use in a generated graph. The figure-1 shows the file menu where tools are available for use in the generated figure plot. These can be used to make the graph effective presentation aids and provide looks as required. One of the useful tools for manipulation of graph insertion of data points in the plot. It can be done by using the Tools option in the file menu bar. In the tools option there is a button Data Cursor which allows the section of a point where the values of axes can be obtained and notified.

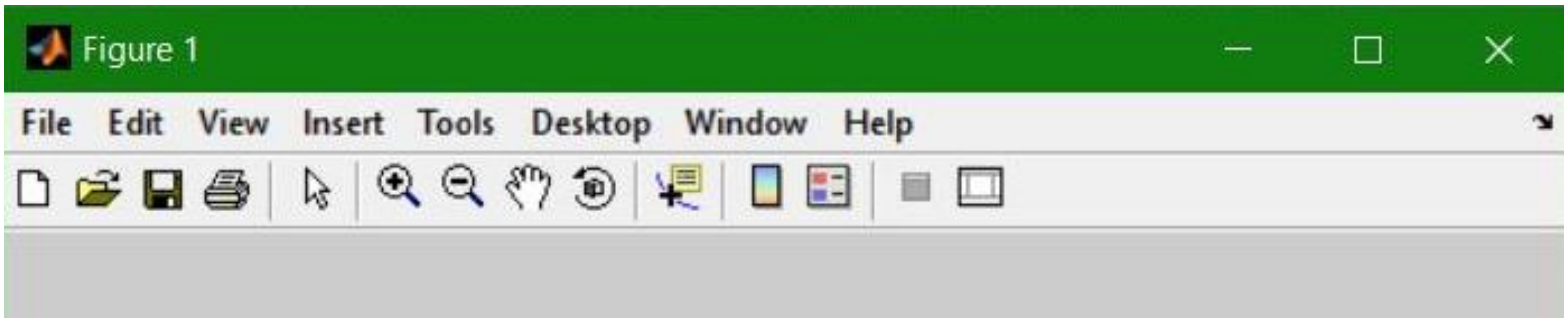


Figure: figure-1

One of the other tools for modifying existing plot is plot editor. To activate this tool, click on the left-leaning arrow in the menu bar. Now we can select and double click on any object in the current plot to edit it.

Overlay Plots:

There are three different ways of generating overlay plots with MATLAB; the plot, hold and line commands.

Method:1 Using the plot command to generate overlay plots

If the entire set of data is available, plot command with multiple arguments may be used to generate an overlay plot. For example, if we have three sets of data-$(x_1, y_1), (x_2, y_2), (x_3, y_3)$- the command to plot overlay is: plot(x_1, y_1, x_2, y_2,':', x_3, y_3,'o') plots(x_1, y_1) with solid line, (x_2, y_2) with a dotted line, and (x_3, y_3) as unconnected points marked by small circles (o).

Method 2: Using the hold command to generate overlay plots
Another way of making overlay plots is with the hold command. Invoking hold on at any point during a session freezes the current plot in the graphics window. All the subsequent plots generated by the plot command are simply added to the existing plot
Method 3: Using the line command to generate overlay plots
The line is low-level graphics command used by the plot command to generate lines. Once plot exists in the graphics window, additional lines may be added by using the line command directly. The line takes a pair of vectors (or a triplet in 3-D) followed by parameter name/parameter value pairs as arguments:

line(xdata,ydata,parameterName, parameterValue)

Sub-Dividing a Figure Window

While dealing with multiple graphs at the same time, it becomes easy to project the plots within the same figure window. It requires a sub-division of the figure window so that different segments of the figure window show different plots. For that MATLAB provides an option called subplot(m,n,o) with m and n represents total number of sub-divisions formed out of the figure window the count of which is kept by o. These two variables m and n effectively can be considered to be like a matrix of m rows and n columns.

The following commands gives four curves in same figure window.

x=-2:0.01:2;

$y = x; y1 = sin(x); y2 = 1./x; y3 = exp(x);$

subplot(2,2,1), plot(x,y)

subplot(2,2,2), plot(x,y1)

subplot(2,2,3), plot(x,y2)

subplot(2,2,4), plot(x,y3)

Polar Plotting

In mathematics, the polar coordinate system is a two dimensional coordinate system in which each point on a plane is determined by a distance from a reference point and an angle from a reference direction. The command [theta, rho] = cart2pol(x,y) transforms corresponding elements of the two-dimensional Cartesian coordinate arrays x and y into polar coordinates theta and rho. The command [theta,rho,z] = cart2pol(x,y,z) transforms three-dimensional Cartesian coordinate arrays x, y, and z into cylindrical coordinates theta, rho, and z. Similarly the command [x, y] = pol2cart(theta,rho) transforms corresponding elements of the polar coordinate arrays theta and rho to two-dimensional Cartesian coordinates and the command [x,y,z] = pol2cart(theta,rho,z) transforms corresponding elements of the cylindrical coordinate arrays theta, rho, and z to three-dimensional Cartesian, or xyz, coordinates.

The syntax polar(theta, rho) makes a plot using polar co-ordinates of the angle theta, versus the radians rho.

There are many specilized graphics functions for 2-D plotting. They are used as alternatives to the plot command. Here, we provide a list of other functions commonly used for plotting x-y data

area	creates a filled area plot
bar	creates a bar graph
barh	creates a horizontal bar graph
comet	makes an animated 2-D plot
compass	creates arrow graph for comples numbers
contour	makes a contour plots
contourf	makes filled contour plots
errorbar	plots a graph and puts error bars
feather	makes a feather plot
fill	draws filled polygons of specified colors
hist	makes histograms
pie	creates a pie chart
stem	plots a stem graph

3-D Plotting

A function named **plot3()** can be used to generate 3-D plots in MATLAB. If the function is used as plot3(x,y,z), the vectors are plotted against the rows or columns of the matrix, depending whether the vectors lengths equal the number of rows or the number of columns. If the lengths are different an error will be created.

Suface Plot

At times a programmer has to deal with ordered pairs, i.e., data that is dependent on both x and y values as z=f(x,y). This can be done by computing a z value for each x,y pair which in effect is to iterate through a nested loop, but one of the major advantages of MATLAB is that it can deal with matrices without resorting to looping. If the data available is in a matrix format this can be done easily. If data is assigned to x and y considered to be vectors, MATLAB provides a useful function called meshgrid that can be used to simplify the generation of X and Y matrix arrays used in 3-D plots.

It is invoked using the form [X,Y]=meshgrid(x,y), where x and y are vectors that help specify the region in which co-ordinates, defined by element pairs of the matrices X and Y, will lie. The matrix X will contain replicated rows of the vector X, while Y will contain replicated columns of vector y.

One of the basic surface plotting is **mesh**. The syntax is **mesh(X,Y,Z)**. mesh(X,Y,Z) draws a wireframe mesh with color determined by Z so color is proportional to surface height. If X and Y are vectors, length(X) = n and length(Y) = m, where [m,n] = size(Z). In this case, (X(j), Y(i), Z(i,j)) are the intersections of the wireframe grid lines; X and Y correspond to the columns and rows of Z, respectively. If X and Y are matrices, (X(i,j), Y(i,j), Z(i,j)) are the intersections of the wireframe grid lines. Another basic surface plotting is surf. The syntax is **surf(X,Y,Z)**. On the following pages we show the examples of these commands

Plot $z = x^2 + y^2$ in $-3 \leq x, y \leq 3$

Plot $z = x^2 + y^2$ in $-3 \leq x, y \leq 3$

```
>> x=-3:0.1:3;y=x;
```

Plot $z = x^2 + y^2$ in $-3 \leq x, y \leq 3$

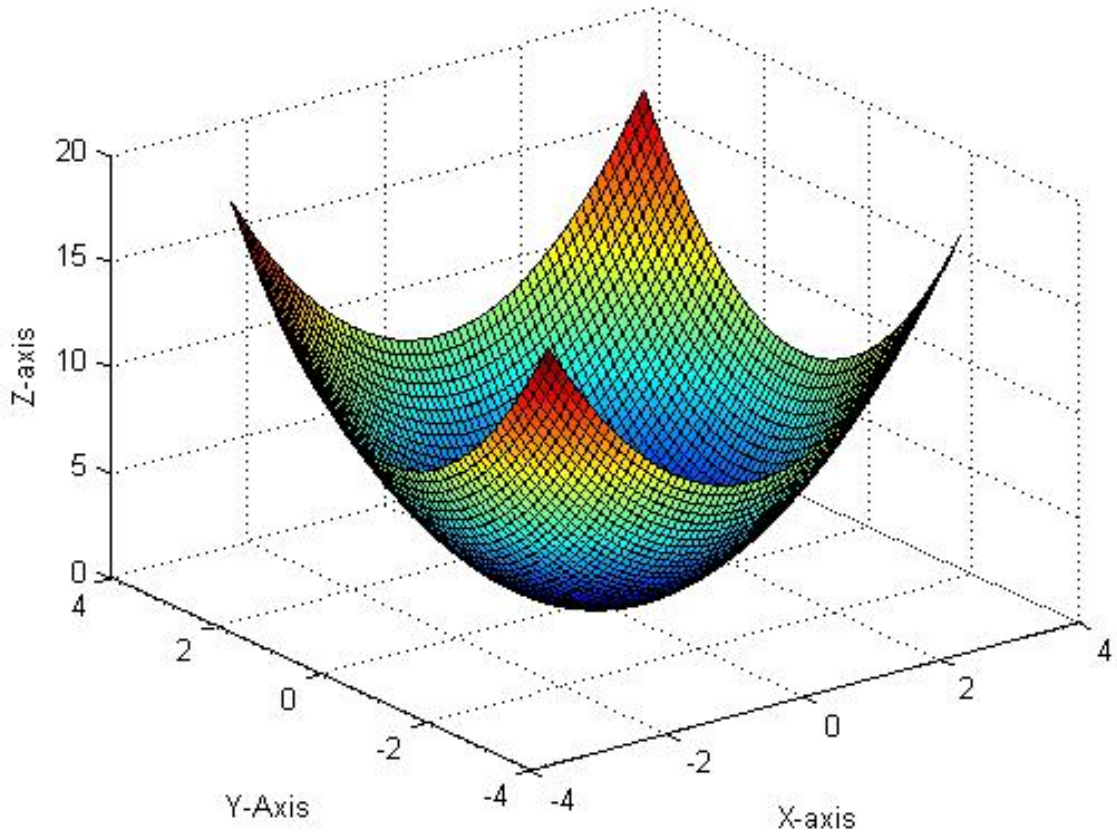
```
>> x=-3:0.1:3;y=x;
>> [x y]=meshgrid(x,y)
```

Plot $z = x^2 + y^2$ in $-3 \leq x, y \leq 3$

```
>> x=-3:0.1:3;y=x;
>> [x y]=meshgrid(x,y)
>> z = x.^2 + y.^2
```

Plot $z = x^2 + y^2$ in $-3 \leq x, y \leq 3$

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



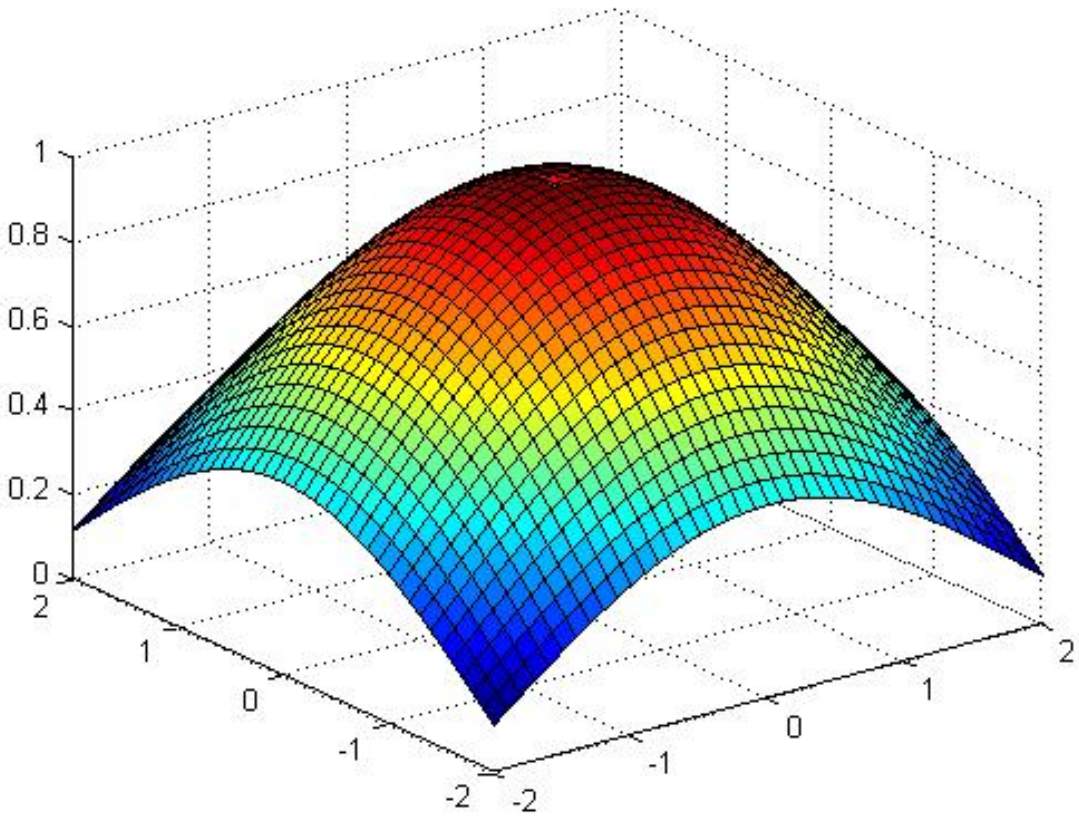
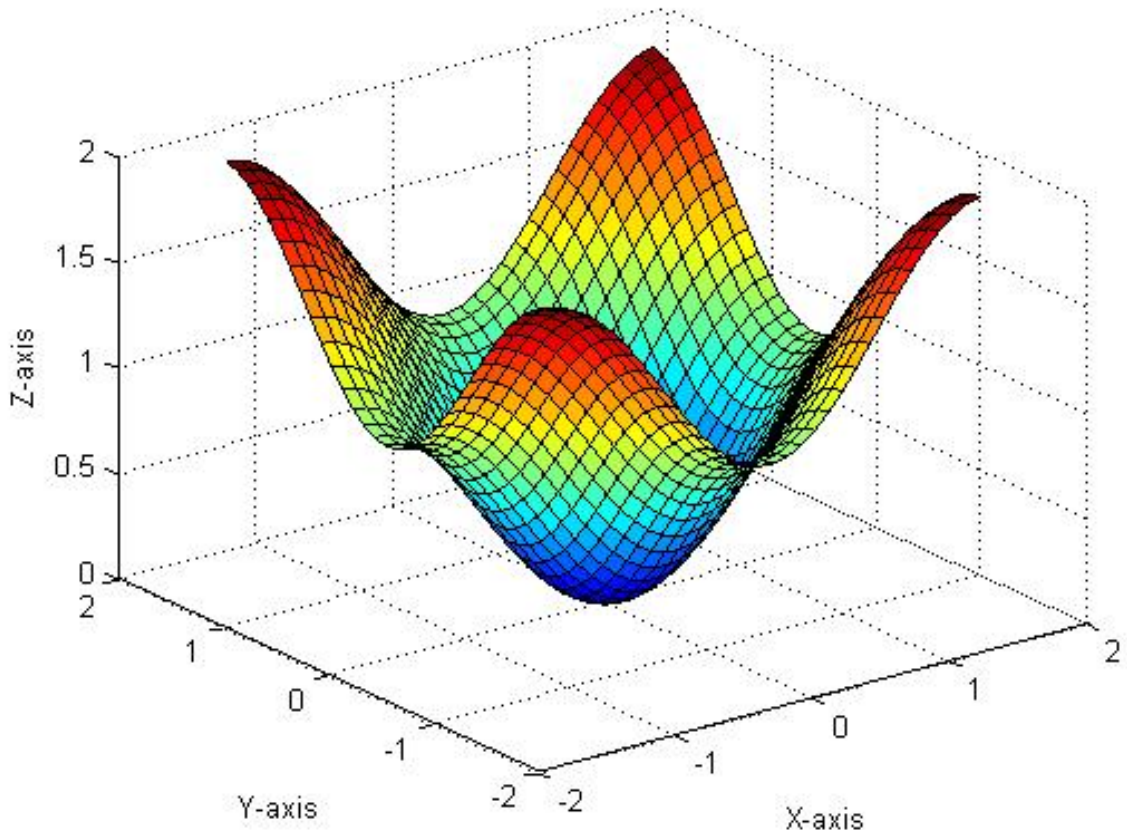
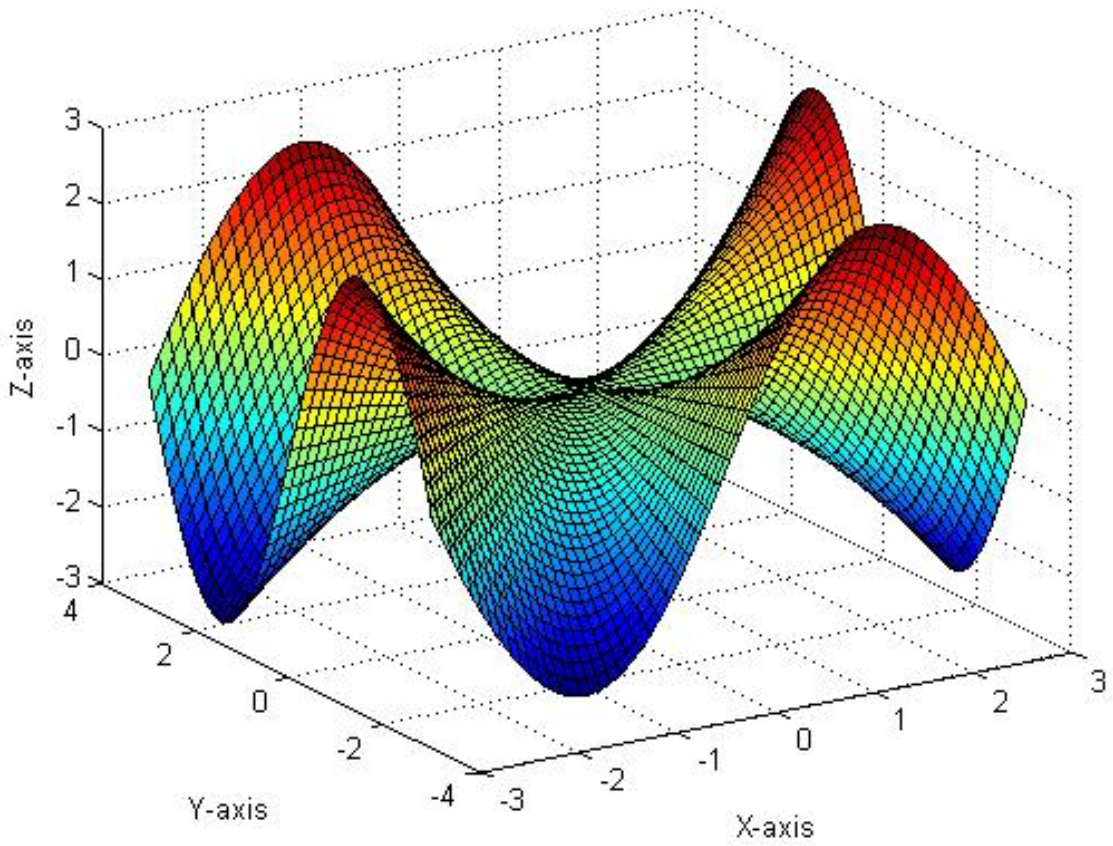


Figure: $z = \frac{sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$,$-2 \leq x, y \leq 2$

$z = sin^2x + sin^2y,\ -\pi/2 \leq x, y \leq \pi/2$



$$z = \frac{xy(x^2 - y^2)}{x^2 + y^2}, \; -3 \leq x, y \leq 3$$



Thank You!