Graphics

By

Dr. R. R. Manza

Objectives

- Introduction
- 2-D Plotting
- Plot Style Options
- Edit Plot
- Basic Statistics of the Graph
- The Plots Creating
- Animation
- Graphics Object Handling
- 3-D Visualization
- Summary

Introduction

This chapter introduces 2-dimensional and 3-dimensional visualization, plotting, overview of specialized plotting, handle graphics object, visualization technique, basic plotting and formatting of plotting etc. MATLAB has several high-level graphical routines. They allow a user to create various graphical objects including two- and three-dimensional graphs, graphical user interfaces (GUIs), movies, to mention the most important ones.

MATLAB provides in-built functions to extensive plot capabilities like plotting data, editing plots, change the view of plots, add styles, titles, colors etc. Using these functions you can build 2-dimensional and 3-dimensional plots easily. MATLAB also provides the commands to manipulate or customize the graphs.

2-D Plotting

Some basic steps for plotting the 2-Dimensional data:

- Firstly select your data, which you want to plot.
- Select figure or position to plot region within a window (optional step). By default MATLAB set the position of plot.
- Using plot function you can plot the data.
- You set the properties like line grid, axis limit, style, width etc.
- You can also give title, assign the labels to axis and drown legend in plot.

Cont..

Some functions of basic plotting

Function	Syntax	Description
Plot	>>plot(x,y,'style') x,y is input parameter	Plot the 2-D simple graph for both axis
loglog	>>loglog(x,y,'style')	Plot with logarithmic scales for two axes.
semilogx	>>semilogx(x,y,'style')	Graph with logarithmic scales for the x-axis and linear scale for the y-axis
semilogy	>>semilogy(x,y,'style')	Graph with logarithmic scales for y-axis and linear scales for x-axis
Plotyy	>>plotyy(x1,y1,x2,y2)	Graph with y tick labels on the left and right side

Plot Style Options

Various line types, plot symbols and colors may be obtained with plot(x,y,'style')

Color Style:

Color	MATLAB Notation
Black	k
White	w
Blue	b
Green	g
Red	r
Cyan	c
Magenta	m
Yellow	y

Cont..

Line Styles

Line Style	MATLAB Notation
Solid	- (default)
Dashed	--
Dotted	:
Dash-dot	-.
No line	none

Cont..

Marker Styles:

Marker Style	MATLAB Notation
Plus sign	+
Circle	O
Asterisk	*
x-mark	x
Point	.
Square	s
Diamond	d
triangle (down)	v
triangle (up)	^
triangle (left)	<
triangle (right)	>
pentagram	p
hexagram	h

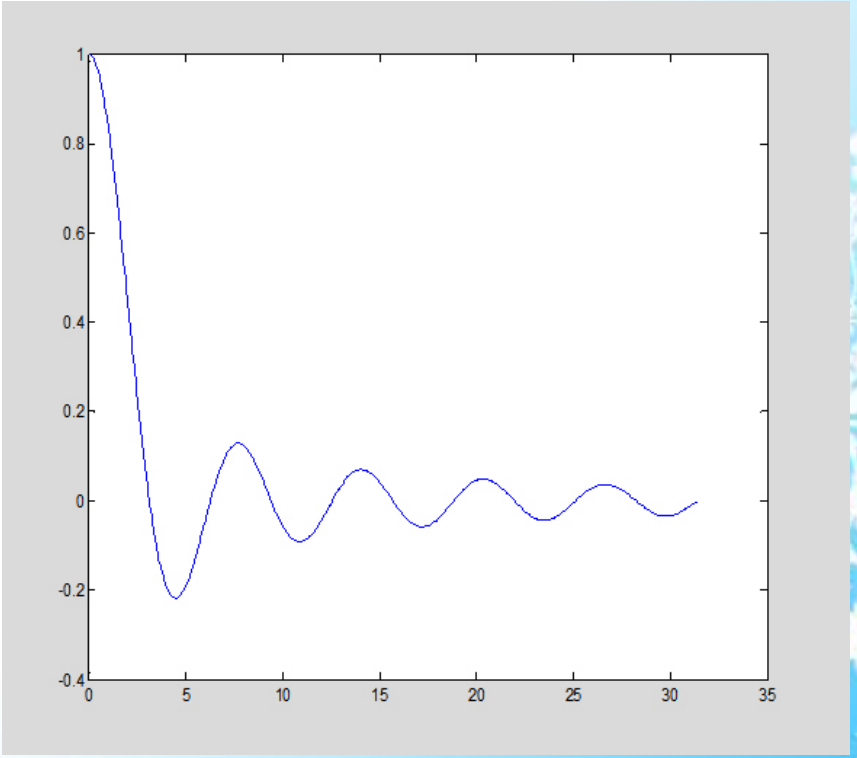
Cont..

The line style and color uses by default for plotting a plot in MATLAB. You can also change the style and color of the graph. By using editing plot you can add annotations to the graph. You can edit graph using the command line. Use the functions set() and get() to set and get the properties of the object in the graph. And you can also use the point and click editing to change the format of the object by double clicking on the object. In this mode you access the properties through the graphical user interface called the property editor. You need to be using a combination of interactive editing and command line editing to get the desired effect.

Cont..

Example 1: Simple plot function

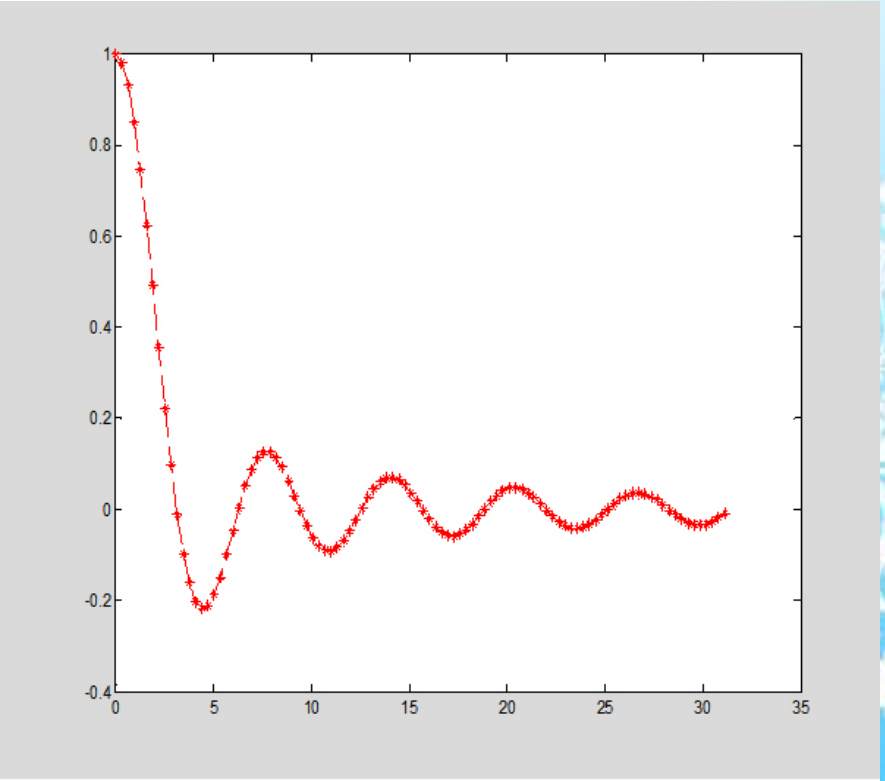
```
x = pi/100:pi/100:10*pi;
y = sin(x)./x;
plot(x,y)
```



Cont..

Example 2: Plot Graph with option

```
x = pi/100:pi/10:10*pi;
y = sin(x)./x;
plot(x,y,'r--*')
```

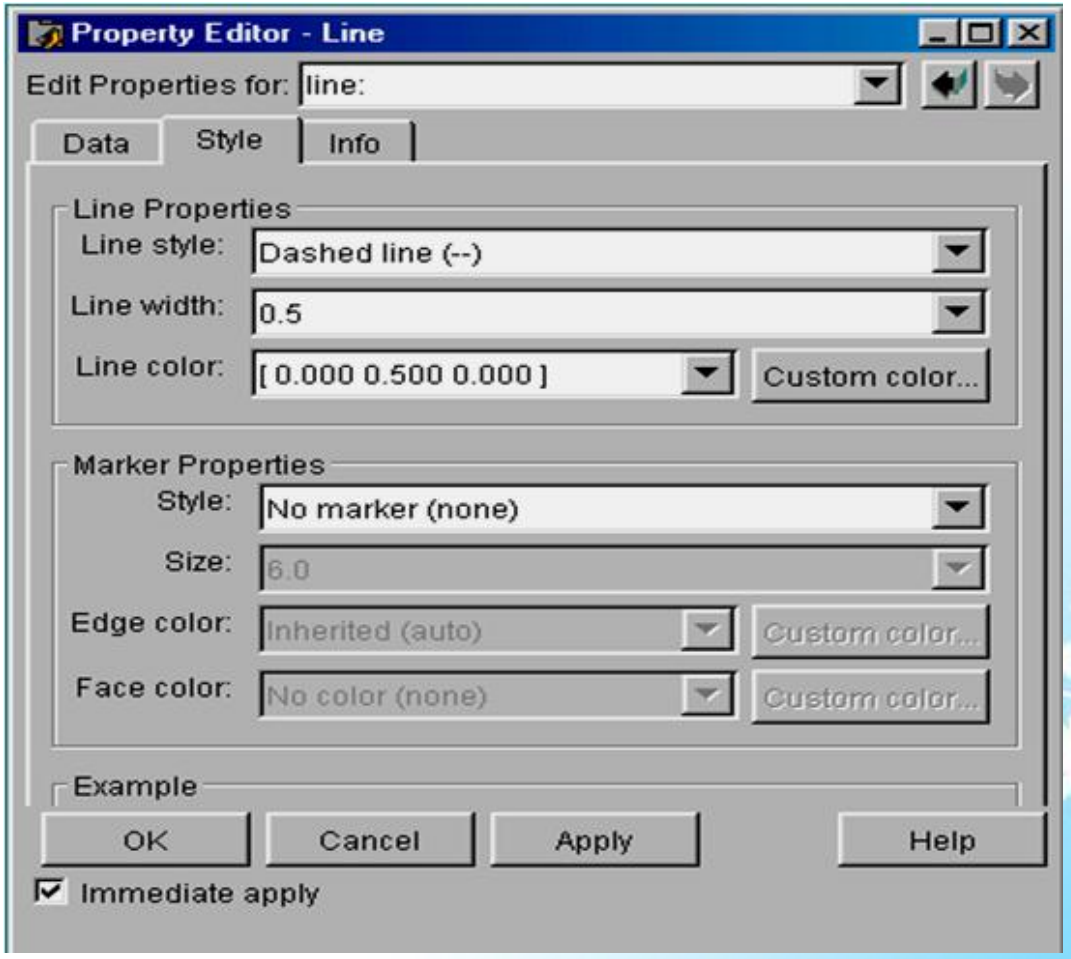


How to Edit Plot

The following steps give instruction to edit a plot:

1) Choose the button to start the plot edit mode or to choose the edit plot option from the tools menu. For exit from the plot edit mode, do the same procedure.
2) Select object in the graph: start the plot edit mode and move the cursor over the object and click on it. For choosing multiple objects you can use the shift + click.
3) Cutting, Copying, and Pasting objects: select the object you want to cut copy or paste and then select the option from the edit menu or you can also use the standard short cuts or also use the right click on the selected object and set the parameter which you want.
4) You also cut the object in one figure to paste another figure. You can also save the figure. MATLAB stores it as the figure with .fig extension. Select save option of file menu from figure window and click the save button. Specify the name which you want to give to the figure file. To open the figure file select open option of file menu from the figure window and select the figure file which you want to open and click ok.
5) If you want to delete any object in the graph, select the object and select clear option from the edit menu or right click on selected button and click on clear or also use windows standard shortcuts.
6) By using the property editor you can change the style of lines, edit title, add labels or edit labels, create or edit legend, add lines, arrows and text in the graph. To start the property editor by using the double click on the figure or select axis properties, figure properties or object properties from the edit menu of the figure window.

Cont..


Property Editor - Line
Edit Properties for: line:
Data
Style
Info
Line Properties
Line style: Dashed line (--)
Line width: 0.5
Line color: [0.000 0.500 0.000]
Custom color...
Marker Properties
Style: No marker (none)
Size: 6.0
Edge color: Inherited (auto)
Custom color...
Face color: No color (none)
Custom color...
Example
OK
Cancel
Apply
Help
Immediate apply

Cont..

Example:

```
>>x=linspace(0,2*pi);
>> y=sin(x);
>> plot(x,y);
>> y1=x;
>> plot(x,y,x,y1,'--');
```

Cont..

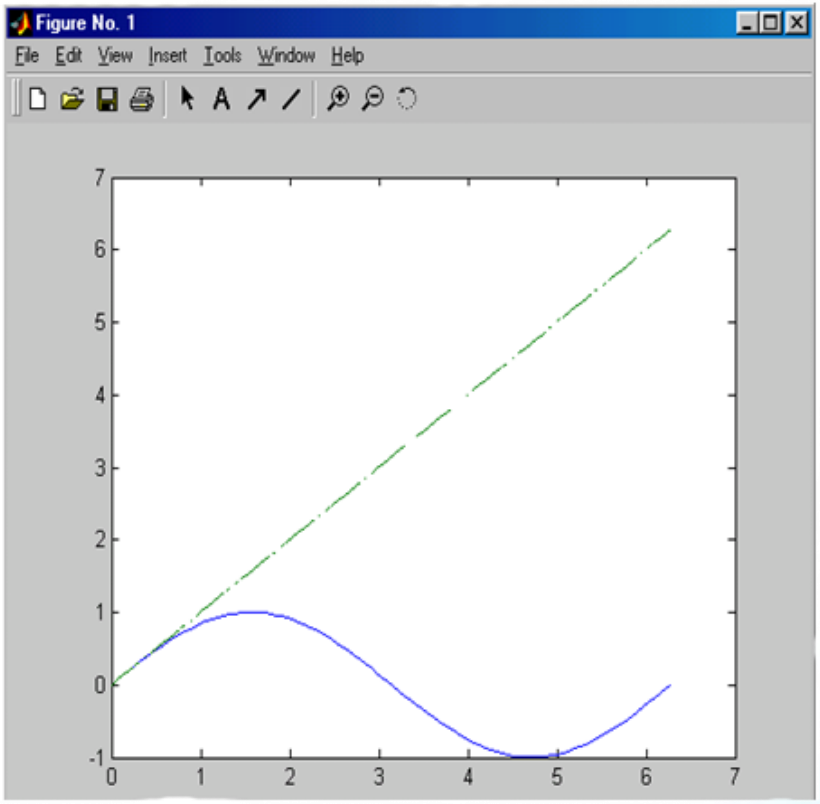
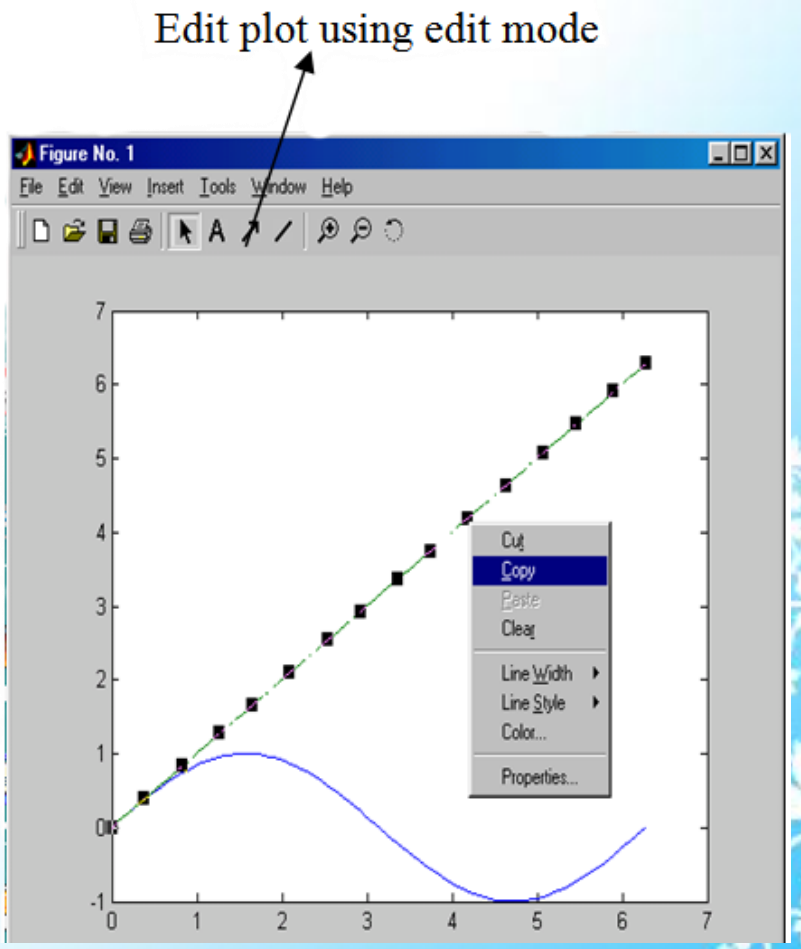


Figure No. 1
File Edit View Insert Tools Window Help


Edit plot using edit mode
Figure No. 1
File Edit View Insert Tools Window Help
Cut
Copy
Paste
Clear
Line Width
Line Style
Color...
Properties...

Cont..

Title:

You can add title in three different ways:

- You can add the title by using the command line and function title on the command prompt. The title function specifies the value of title properties at the time when you create plot.

  Syntax: - title ('enter your plot title'); %Ex. title ('sign function');

- Add title by using property editor: To start property editor, double click on the axes in the graph. You can also start property editor by right click on the axes and select the properties and then select the label and add the title which you want to give and click on apply.
- Add title by using insert menu: Click the insert menu in the figure window menu bar and choose title. MATLAB opens a text entry box, enter the title and click any where in the figure.

Cont..

Legend:

You can create or add the legend in two ways:

- Create legend by using command line: You can create legend to a graph by using the legend command on command prompt but you must specify the text labels when you create a legend function. You can also specify its many aspects like position.

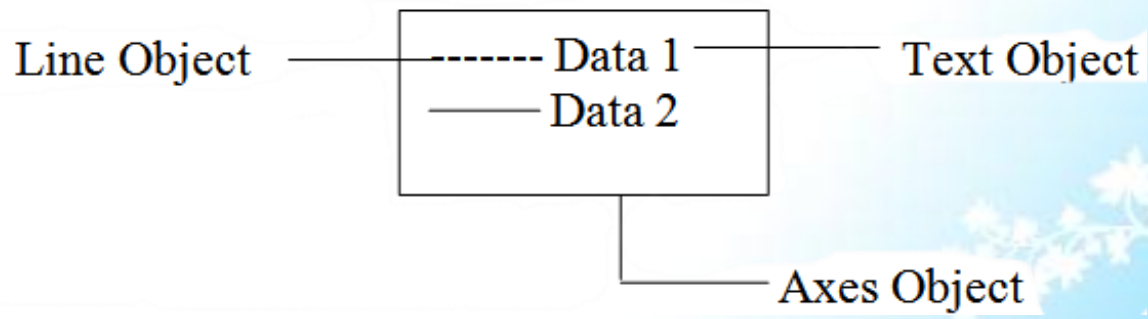
  Syntax: - legend (linestyle1, string1…) or legend (string1, string2…)

  Example: legend ('linear','sin x');

- Create legend by using Insert Menu: You can also create the legend by using the insert menu, click the insert menu and choose the legend option. MATLAB automatically create a legend and place it in the upper right corner of the graph.

Cont..

Legend is created by a separate axes object or box. You can move this object anywhere in the axes. This object contains one or more lines and text representing the sample of plot and labels of data set respectively plotted of the graph. When you edit or resize a legend first start plot in edit mode. The following figure shows the structure of the legend.



Cont..

Axes Labels:

You can enter labels in three different ways:

- Add label option using Insert menu: Do the same procedure as add titles. Click the insert menu and choose the xlabel, ylabel or zlabel and enter the label in the text entry box and click anywhere in the figure window.

  Syntax: xlabel(' x axis'); %Ex. xlabel('→ x');
  ylabel(' y axis'); %Ex. ylabel('approximation of sin');
  zlabel(' z axis');

- Add label using the property editor: First start the property editor using double click on the axes in the graph; when the properties appear select the labels, enter the text of the label in text entry box and click on apply button.
- Add labels using the command prompt: You can add the x, y, and z axis labels by using commands xlabel, ylabel, and zlabel commands respectively. MATLAB automatically sets the position of text on the corresponding axis.

Cont..

Text annotation:

You can add the text annotation anywhere in the figure window either using command prompt or using the plot edit mode in figure window toolbar. You can use the gtext or text commands for adding text to graph. Text command is used when you want to position a text annotation according to the point of data set. When you use gtext command enter your text annotation and press enter or return, move cursor on graph and push mouse button where you want to place text annotation. You can also edit the text annotation; firstly start the graph in edit mode and select the text annotation object to move anywhere in the graph.

Syntax: >>text(position, 'enter text'); %Ex. text(3,0,'sin(x)');

>> gtext('enter text') %ex. gtext ('linear approximation');

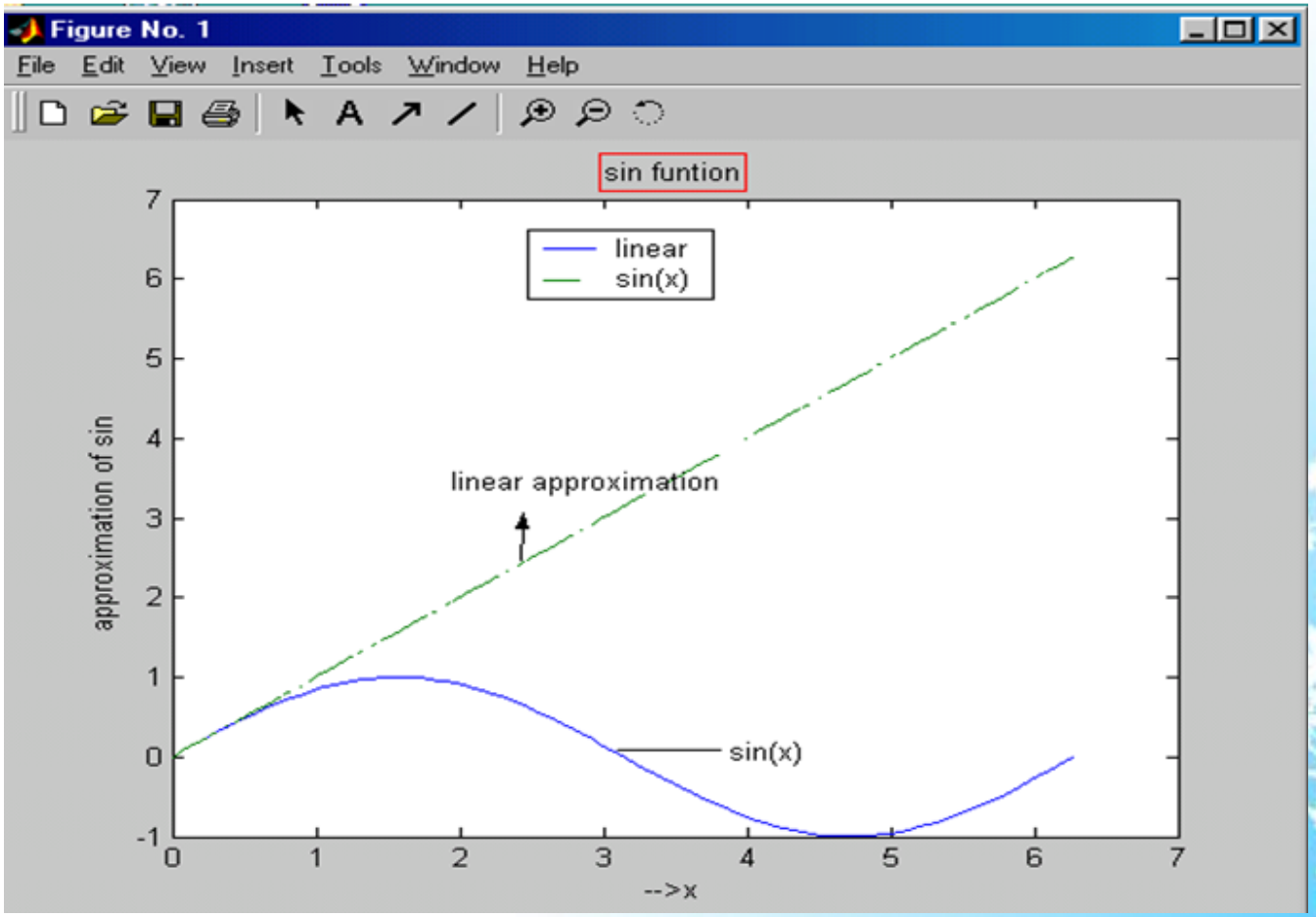
Cont..

Arrows and lines

You can also draw the arrows anywhere in the figure window. Firstly start the plot in plot editing mode, then click the arrow or line button in figure window toolbar or select arrow or line in insert menu. Move cursor at position where you want to draw arrow or set the cursor position in the figure. Hold the mouse button down and move the mouse to define the length and direction of the line or arrow. You can edit the arrow or line using the property editor. You can change the style, width, color and other characteristics of the arrow or line annotation using the fields in the panel tab.

Cont..

All of the above operations apply on the given example below.



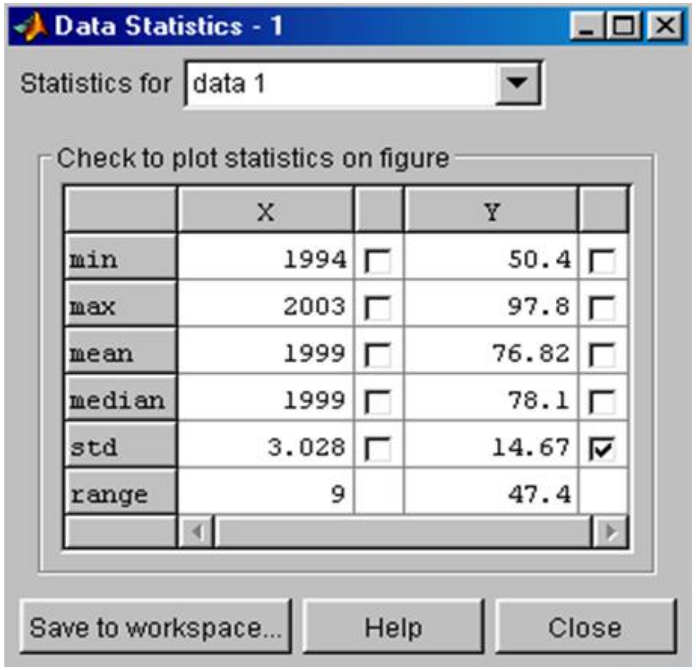
Basic Statistics of the Graph

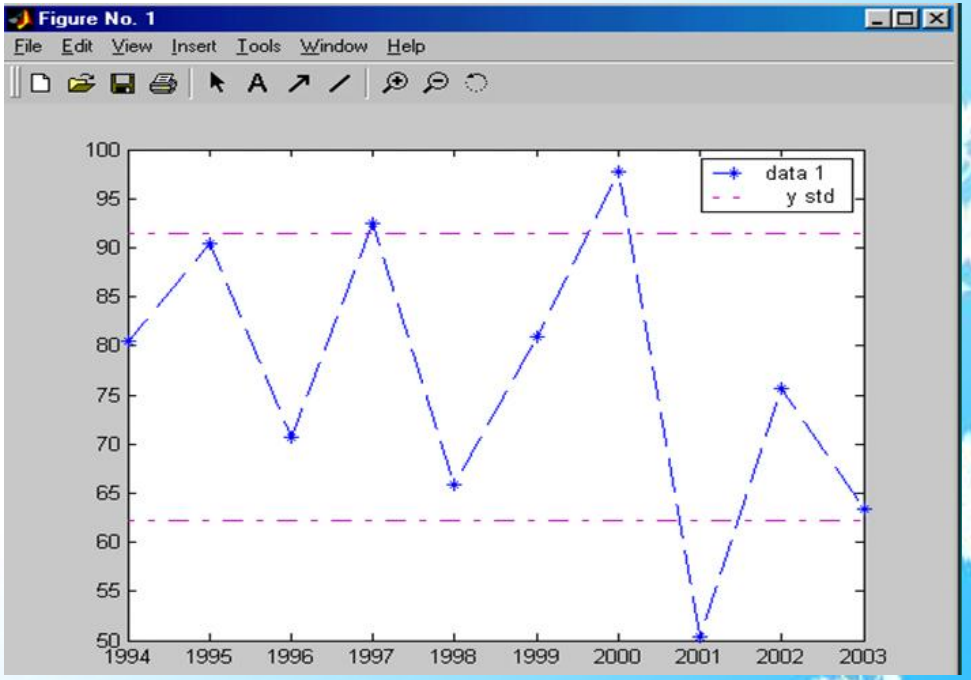
MATLAB data statistic tool calculates basic statistics about the central tendency and variability of data plotted in graph. Data statistic tool will appear for tool menu option from the generated graph window. It also allows to plot these statistics in the generated graph. Data statistic tool provide information about plotting the mean, maximum, minimum, median, range and standard deviation of a data set. To do this MATLAB functions used are mean, max, min, median, n/a, std respectively. Out of these functions for statistics of range in MATLAB direct function is not available but range gives interval between the lowest and highest value in the data set. If the data statistic tool is displayed and we try to change the x data or y data of a plot, the data statistics tool updates the statistics for the plot. We can view multiple plot statistics but it displays the statistics for only one plot at a time. The generated statistics by the data statistic tool can also save to the MATLAB workspace. Another important operation we can do on the graph window is edit graph.

Cont..

Example: Passing percentage of M. Sc. Students since last 10 years.

```
year=[1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 ];
percentage = [80.5 90.4 70.7 92.5 65.9 80.9 97.8 50.4 75.7 63.4];
plot(year,percentage,'--*')
```





The Plots Creating

The user's information can be represented more effectively using various types of graphs. Selection of graph types usually depends on the type of information. Bar and area graphs are useful to represent information of type, time comparing results and to show an individual's contribution in the total amount. Pie Charts are useful to highlight contribution of individuals. Histograms are used to show the distribution of data values. Stem and Stair step plots are used to display discrete data values. To display data of type direction and velocity vectors, compass, feather, and quicker plots are used. Whereas Contour plots show equal values regions of data. Through interactive plotting we can select data points to plot with a pointer. Animations add an addition at data dimension by using sequence plots.

Cont..

Following are the types of graphs

1. Bar Graph

MATLAB supports four types of Bar graphs.

a) Vertical Bar Graph:

b) Horizontal Bar Graph

c) 2-D Bar Graph

d) 3-D Bar Graph

To represent user information in any type of bar graph, the following functions are used:

Cont..

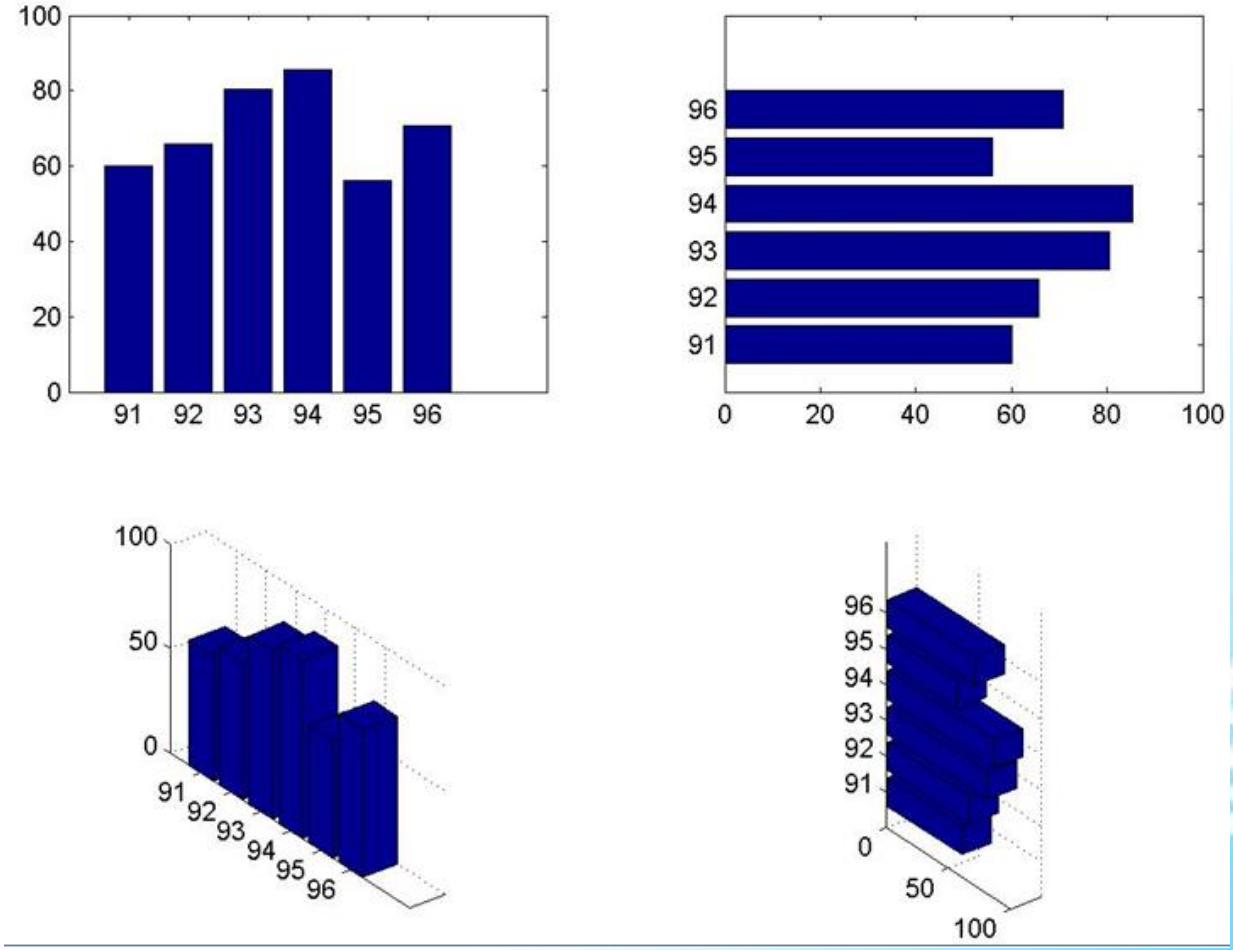
Function	Description
bar	Displays columns of m-by-n matrix as m groups n vertical bars.
barh	Displays columns of m-by-n matrix as m groups n Horizontal bars.
bar3	Displays columns of m-by-n matrix as m groups n vertical 3-D bars.
bar3h	Displays columns of m-by-n matrix as m groups n horizontal 3-D bars.

Cont..

Example:

```
>> per_passing=[60 65.79 80.40 85.44 55.99 70.71];
>>year=91:96;
>> subplot(2,2,1),bar(year, per_passing);
>>subplot(2,2,2),barh(year, per_passing);
>>subplot(2,2,3),bar3(year, per_passing);
>>subplot(2,2,4),bar3h(year,per_passing);
```

Cont..


100
80
60
40
20
0
91 92 93 94 95 96
96
95
94
93
92
91
0 20 40 60 80 100
100
50
0
91 92 93 94 95 96
96
95
94
93
92
91
0
50
100

Cont..

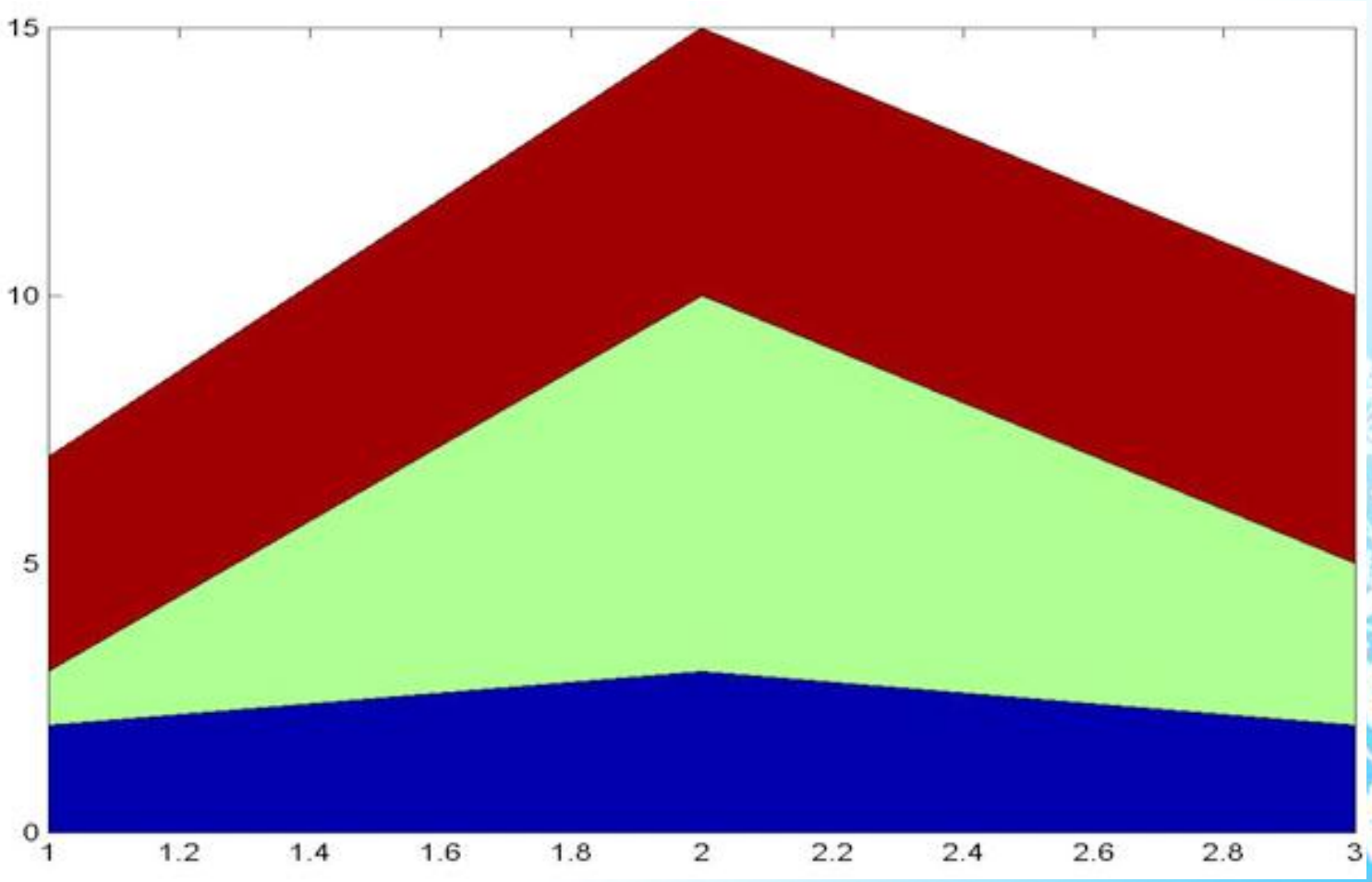
2. Area Graph

Area Graphs are useful for displaying continuous data. Area plots the values in each column of a matrix as a separate curve and files the area between the curve and the x-axis. This type of graph shows how elements in a vector or matrix contribute to the sum of all elements at a particular x location. The height of the area graph is the sum of the elements in each row.

Example:

```
>> y=[2 1 4 ;3 7 5;2 3 5];
>> area(y)
```

Cont..



Cont..

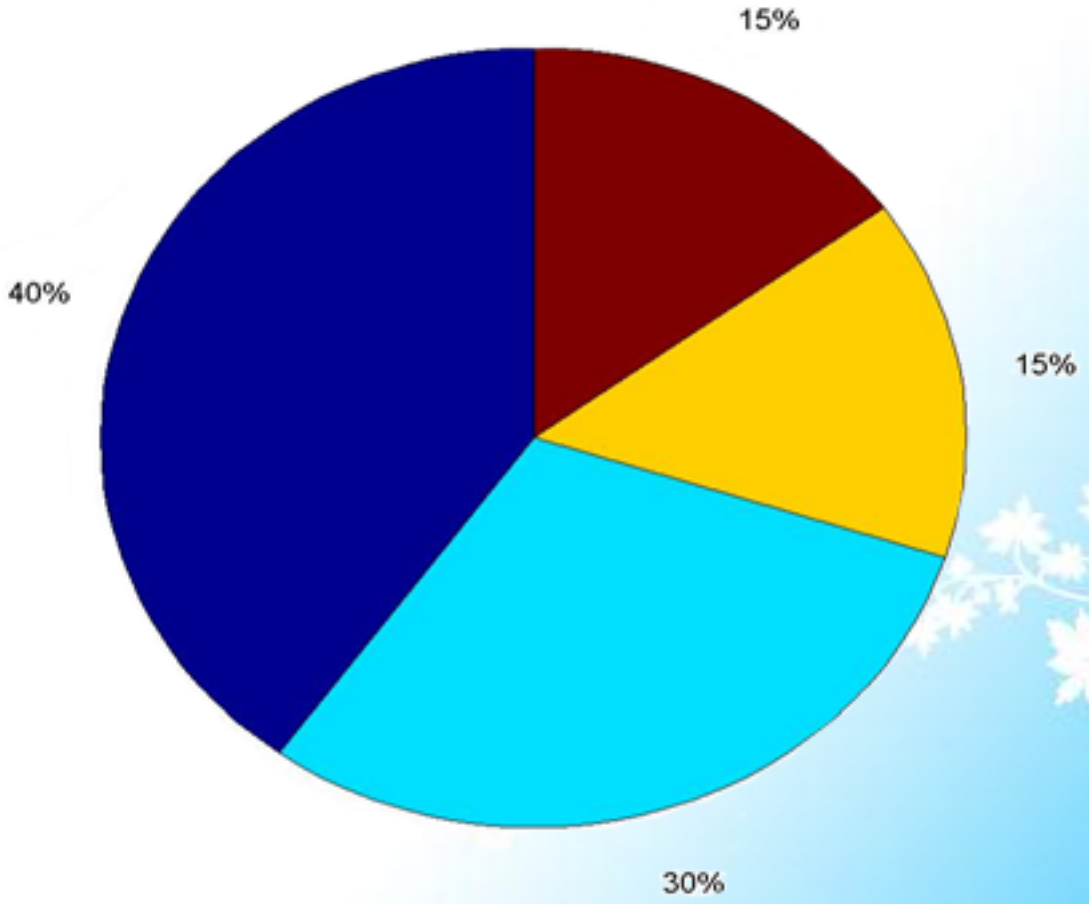
3. Pie Chart

Pie chart displays the percentage that each element in a vector or matrix contributes the sum of all elements. Pie charts can be two dimensional and three dimensional. Pie and pie3 create 2-D and 3-D pie charts.

```
>> v=[40 30 15 15];
>> pie(v)
>> pie3(v)
```

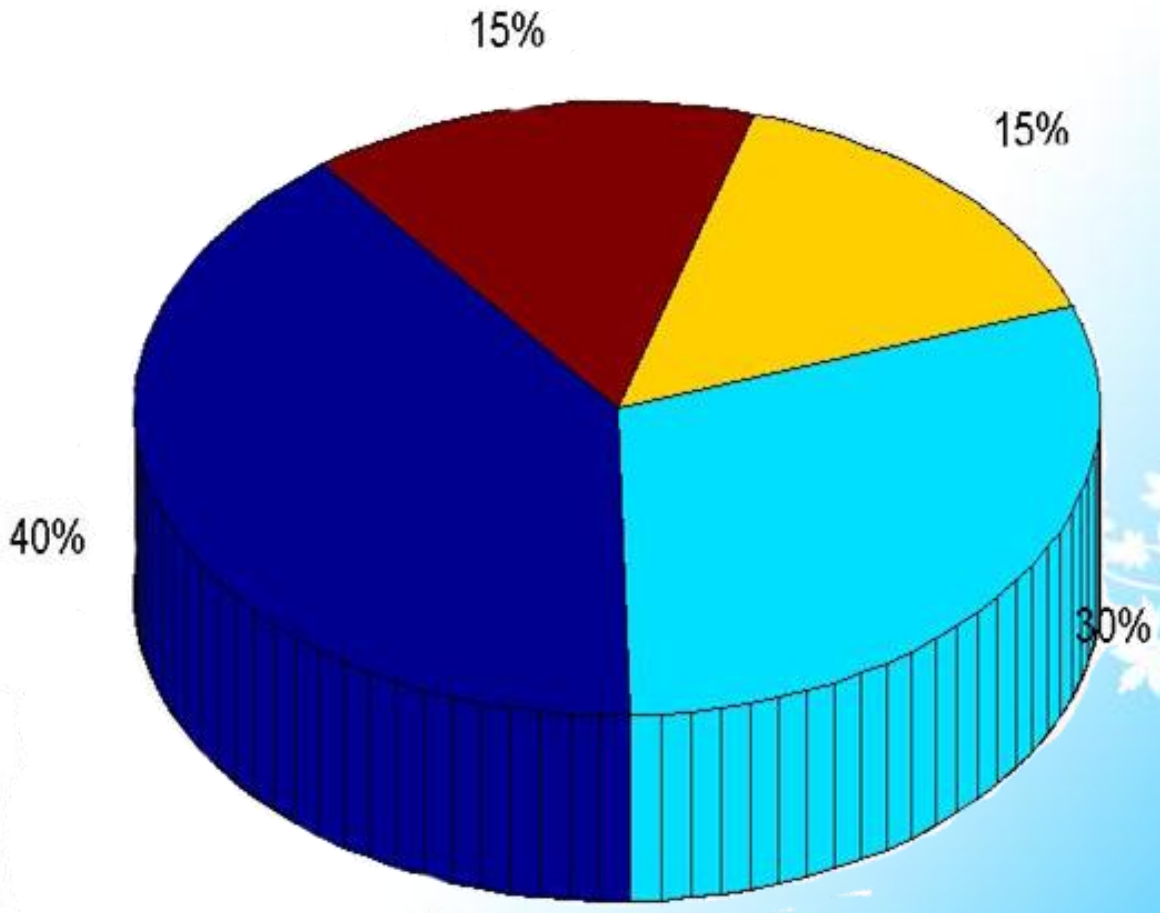
Cont…

2-D Pie Chart



Cont..

3-D Pie Chart



Cont..

4. Histograms

This function shows the distribution of data values. The histogram hist function displays data in a Cartesian coordinate system and rose function displays data in polar coordinate system. The histogram function counts the number of elements within a range and displays each range as a rectangular bin. The height of the bins represents the number of values falls within the range. The hist function shows the distribution of the elements in Y as a histogram with equally spaced bins between the minimum and maximum values in Y. A rose plot displays the data after converting it into radians.

Example:

```
>> h=[2 3 4 2 6 8 6 5 1 10 15 5 4];
>> hist(h)
```

Cont..

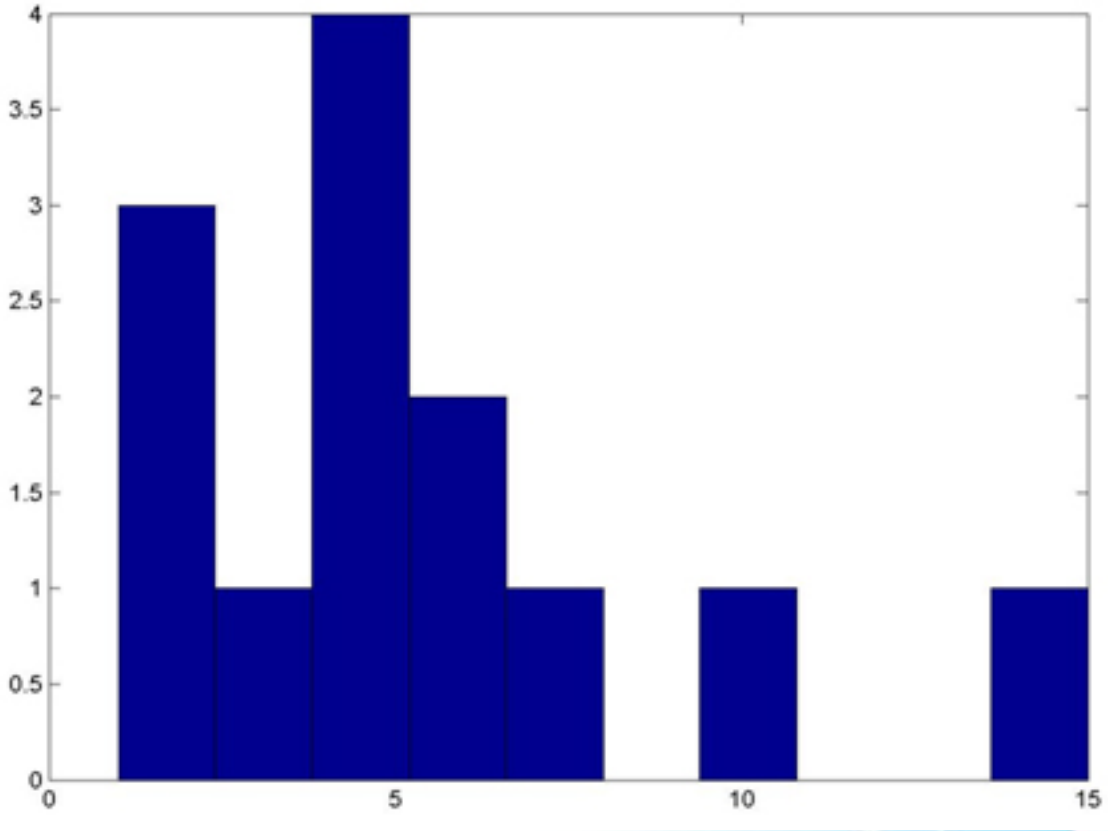


Fig: Cartesian Coordinate System

Cont..

Example:

```
>> h=[2 3 4 2 6 8 6 5 1 10 15 5 4];
>> rose(h)
```

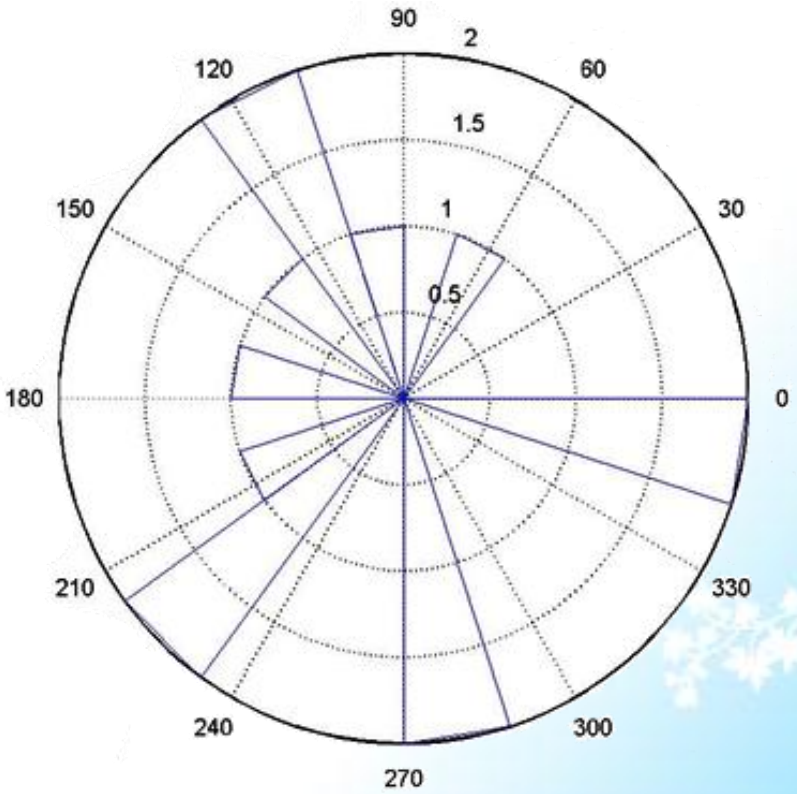


Figure: Polar coordinate System.

Cont..

5. Discrete Data Graph:

These graphs describe how to use stem plots and stair step plots to display discrete data graph, display by function stem, stem3 and stairs. stem displays a discrete sequence of y-data as stems from x-axis. stem3 function displays a discrete sequence of z-data as stems from xy-plane. stairs displays a discrete of y-data as steps from x-axis.

Example: 2-D stem plot.

```
>> temp=[2.4 10.5 3.2 5 7 9.2 1];
>> time=[5 6 7 8 9 10 11];
>> stem(temp, time);
```

Cont..

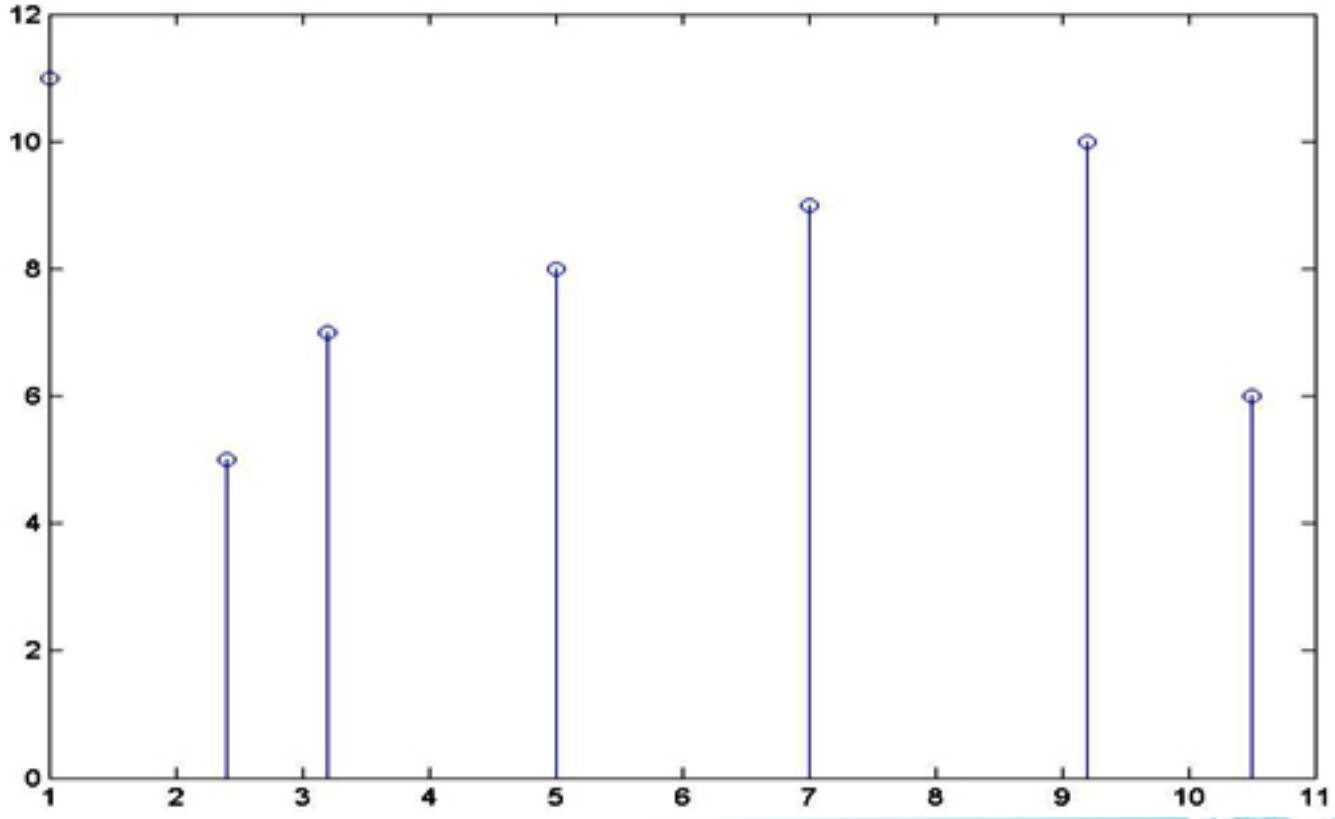


Figure: 2-D stem plot

Cont..

Example: 3-D stem plot

>> stem3(temp,time)

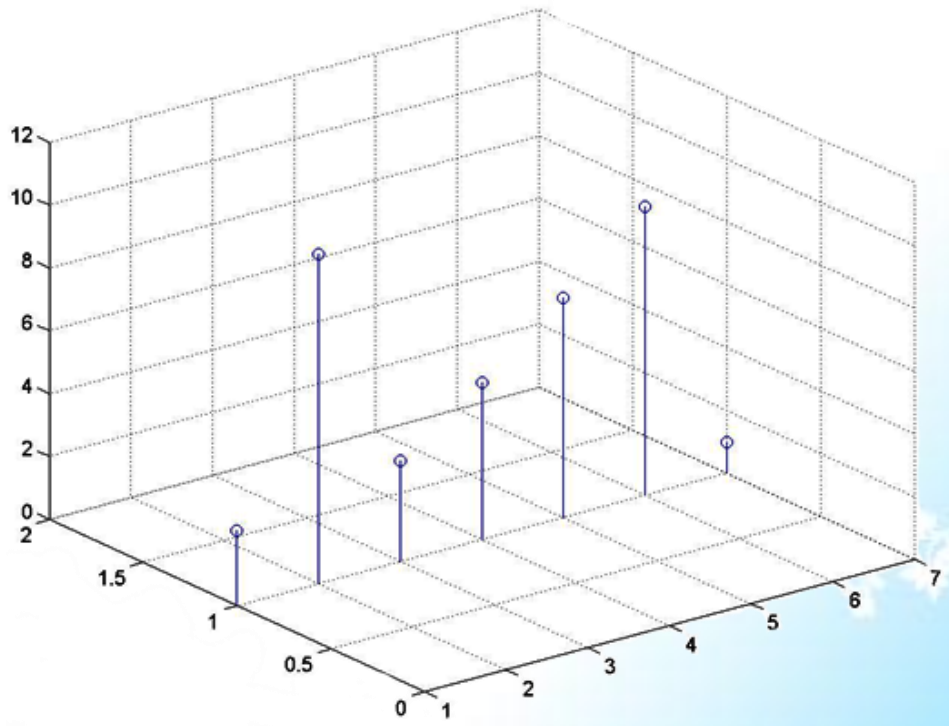


Figure: 3-D stem plot.

Cont..

Example: stair step plot

```
>> i=1:2:30;
>> j=30:-2:1;
>> stairs(i,j)
```

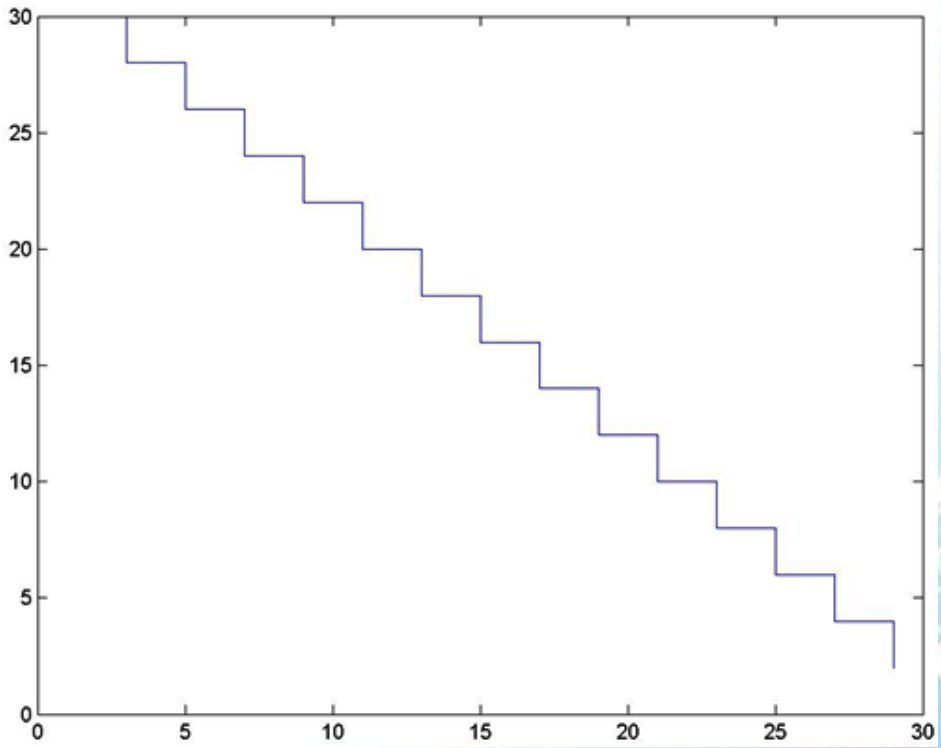


Figure: stair step plot.

Cont..

6. Direction and velocity vector graphs:

This type of graphs displays data consisting of direction vectors and velocity vectors. Following are the vector plotting commands:

Function	Description
compass	Displays vectors emanating from the origin of a polar plot.
feather	Displays vectors extending from equally spaced points along a horizontal line.
quiver	Displays 2-D vectors specified by (u,v) components
quiver3	Displays 3-D vectors specified by (u,v,w) components.

Cont..

7. Contour plots

The contour functions are used to create, display and label isolines which are determined by one or more than one matrices. Contour plotting commands are as follows:

Function	Description
contour	Displays 2-D isolines generated from values given by a matrix z.
contour3	Displays 3-D isolilnes generated from values given by matrix z.
contourf	Displays a 2-D contour plots and fills the area between the isolines with a solid color.
contourc	Calculate the contour matrix used by the other contour functions.

Cont..

Example:

[x,y,z]=peaks;

>> subplot(2,1,1),contour(x,y,z,20)

>> subplot(2,1,2),contour3(x,y,z,20)

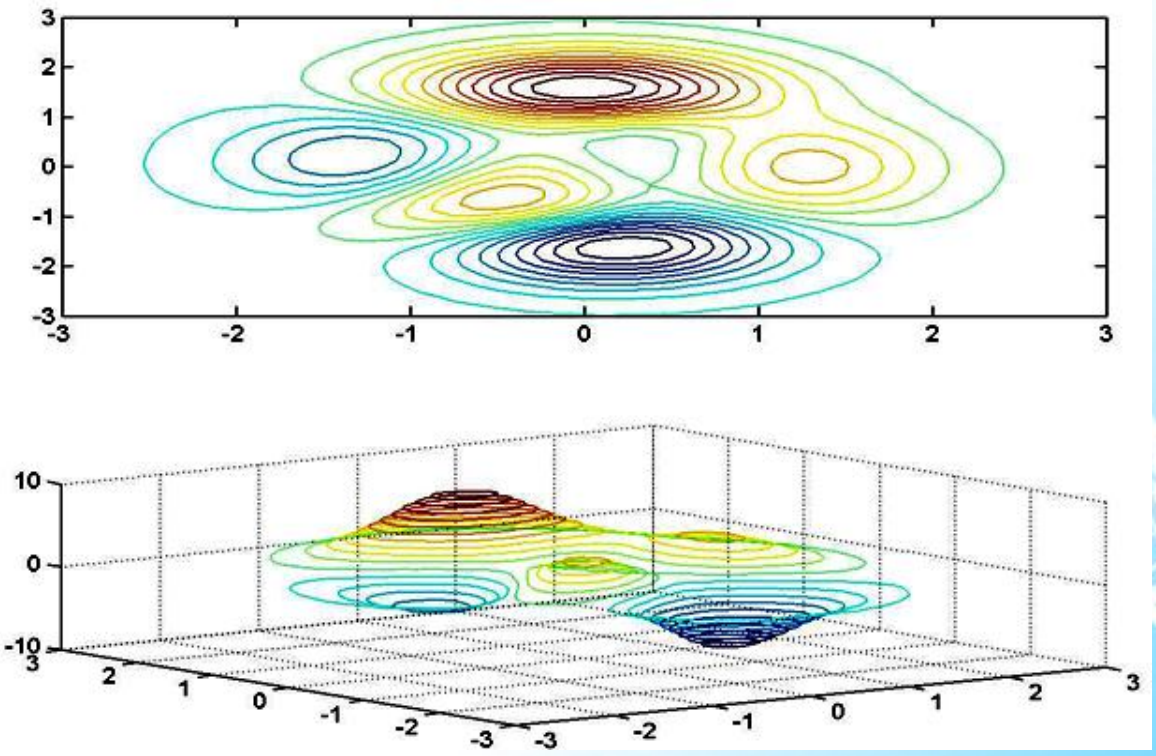


Figure: 2-D and 3-D contour plot.

Cont..

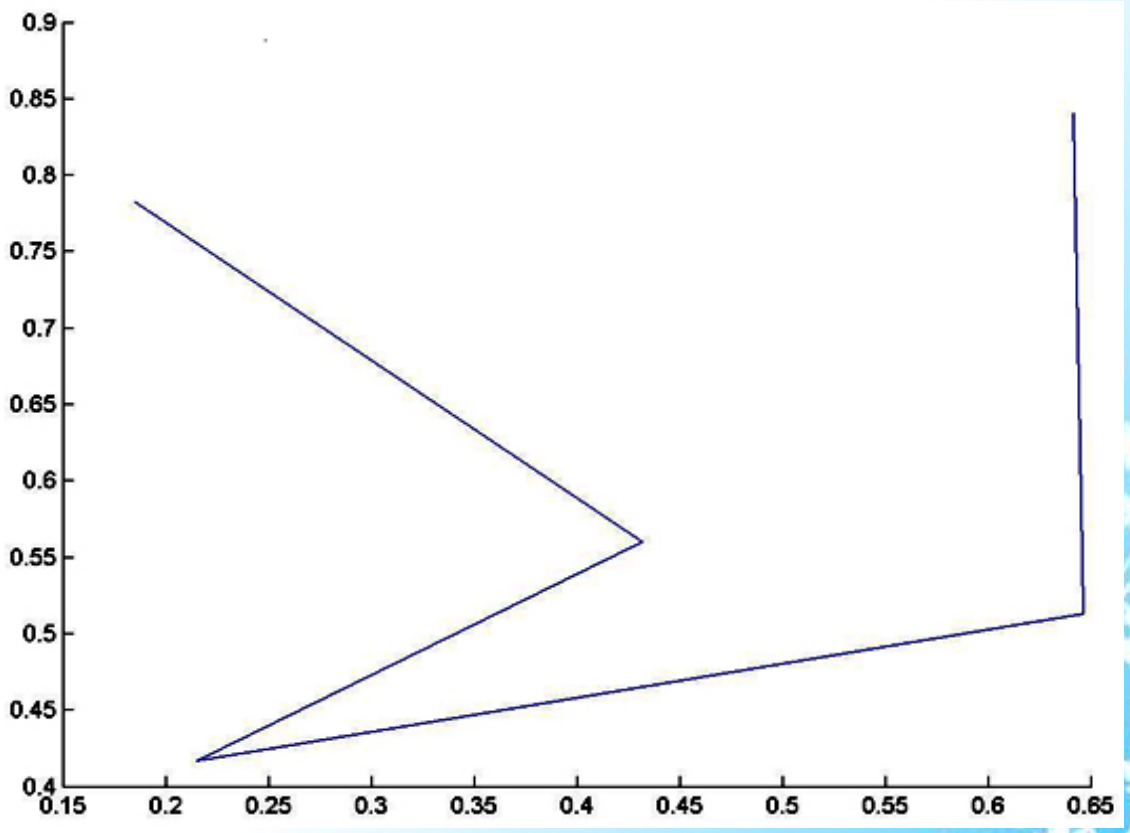
8. Interactive plotting:

Interactive plotting allows plotting user free hand graph. To select the points of free hand graph ginput function is used. ginput function allows to select the mouse or the arrow key to select points to be plotted. This function returns the coordinate of the selected points.

Cont..

Example:

```
>> hold on
>> [x,y]=ginput(5)
x =
    0.1855
    0.4320
    0.2154
    0.6463
    0.6417
y =
    0.7822
    0.5599
    0.4167
    0.5132
    0.8406
>> plot(x,y);
```



Animation

Animated sequences created in MATLAB by two different ways:

1. Saving number of different pictures and then playing them as a movie.
2. Continuously erasing and redrawing the objects and making incremental changes with each redraw.

We can save any sequence of graphs and play the sequence as a movie. This process has two steps:

1. Use getframe to generate each movie frame.
2. Use movie to run the movie for specified number of times and rate.

getframe returns a cdata and colormap as structure of fields.

Cont..

Example:

```
for k=1:10
t=linspace(0,2*pi,200);
r=sqrt(abs(2*sin(5*t)+k));
polar(t,r)
m(k)=getframe;
end
movie(m,5);
```

We can create different effect by selecting the different erase modes. Erase modes are useful for dynamic redrawing. Different erase modes are:

1. none: it does not erase the object when it is moved.
2. background mode erases the object by redrawing it in the background color. This mode erases the object with all below information such as grid lines.
3. xor mode erases only the object and it is usually used for animation.

Cont..

Example:

```
A = [ -8/3 0 0; 0 -10 10; 0 28 -1 ];
y = [35 -10 -7]';
h = 0.01;
p = plot3(y(1),y(2),y(3),'.', ...
 'EraseMode','none','MarkerSize',5);
% Set EraseMode to none
axis([0 50 -25 25 -25 25])
hold on
for i=1:4000
    A(1,3) = y(2);
    A(3,1) = -y(2);
    ydot = A*y;
    y = y + h*ydot;
    % Change coordinates
set(p,'XData',y(1),'YData',y(2),'ZDa
ta',y(3))
    drawnow
    i=i+1;
end
```

Graphics Object Handling

Graphics object handlings are the basic drawing elements used by MATLAB to display data and to create graphical user interfaces. The unique identifier of an object is called handle. This handle can be used to manipulate the object properties or characteristics of an existing graphics object. Graphics objects contain the number of types such as line, text, patch, axes, image, surface etc. Axis objects define a region in a figure window and orient their other information within this region. The different functions create an axis object such as plot, surf, mesh and bar. There are three basic image types such as indexed, intensity and true color. Since images are strictly two dimensional hence we can view them only at the default 2-D. The light object defines the different possible sources of the light that affect all patches and surfaces of the object within the axes. Line object is used to create 2-D and 3D plots. A single patch contains multiple faces each colored independently with solid or interpolated colors. Patch object is created by fill, fill3 and contour3 functions. Text objects are nothing but just the collection of character string. The functions used to handle text objects are title, xlabel, ylabel, zlabel and gtext.

Cont..

Common properties of all objects:

Property	Description
BusyAction	Control the way MATLAB handles callback routine interruption defined for the particular object.
ButtonDownFnc	Callback routine that executes when button press occurs.
Children	Handles of all this object's children objects.
Clipping	Mode that enables or disables clipping
CreateFnc	Callback routine that executes when this type of object is created.
DeleteFnc	Callback routine that executes when you issue a command that destroys the object.
HandleVisibility	Allows you to control the availability of the object handle from the command line and from within callback routines.
Interruptible	Determine whether a callback routine can be interrupted by a subsequently invoked callback routine.
Parent	The objects parent
Selected	Indicates whether object is selected
SelectionHighlight	Specifies whether object visually indicates the selection state.
Tag	User-specified object label.
Type	The type of object ex. figure, line, text, etc.

Cont..

List of Graphics Object Creation Functions :

Function	Description
axes	Rectangular coordinate system that scales and orients axes children image, light, line, patch, surface, and text objects.
Figure	Window for displaying graphics
Image	2-D picture defined by either color map indices or RGB values. The data can be 8-bit or double precision data.
Light	Directional light source located within the axes and affecting patches and surfaces.
Line	Line formed by connecting the coordinate data with straight line segments, in the sequence specified.
Patch	Polygonal shell created by interpreting each column in the coordinate matrices as a separate polygon.
Rectangle	2-D filled area having a shape that can range rectangle to an ellipse.
surface	Surface created with rectangular faces define by interpreting matrix elements as heights above a plane.
text	Character string located in the axes coordinate system.
uicontextmenu	Context menu that you can associate with other graphics object.
Uicontrol	Programmable user interface device such as push button, slider, or list box.
uimenu	Programmable menu appearing at the top of figure window.

Cont..

Property value setting:

The set and get function specify and retrieve the value of existing graphics object properties. These functions are unable to list possible values of fixed set of properties. Properties of an existing object can change using the set function and the handle return by the creating function.

Syntax:

set(object_handle, 'PropertyName', 'NewPropertyValue');

To return current value specific object property :

Syntax

returned_value=get(object_hanle,'PropertyName');

3-D Visualization

Visualization is the process of representing graphical data. It makes certain characteristics or values more apparent. Geometric forms of visualization convey information like surfaces; solids and colors are mapped to data values. The geometric form can represent real life examples such as airplane or wave guide or may be graphical elements that indicate data values, such as streamlines or slice planes. 3-D visualization consist of creating 3-D graphs, defining the view, lighting as a visualization tool, transparency, creating 3-D models with patches, and volume visualization techniques.

Cont..

Example:

```
>> x=[11 22 33 1 3 14 15 18];
>> y=[19 18 17 16 10 11 99 33];
>> z=[22 33 42 51 75 48 97 52];
>> a=[x y z];
>> b=[y x z];
>> c=[z x y];
>> plot3(a,b,c)
```

Cont..

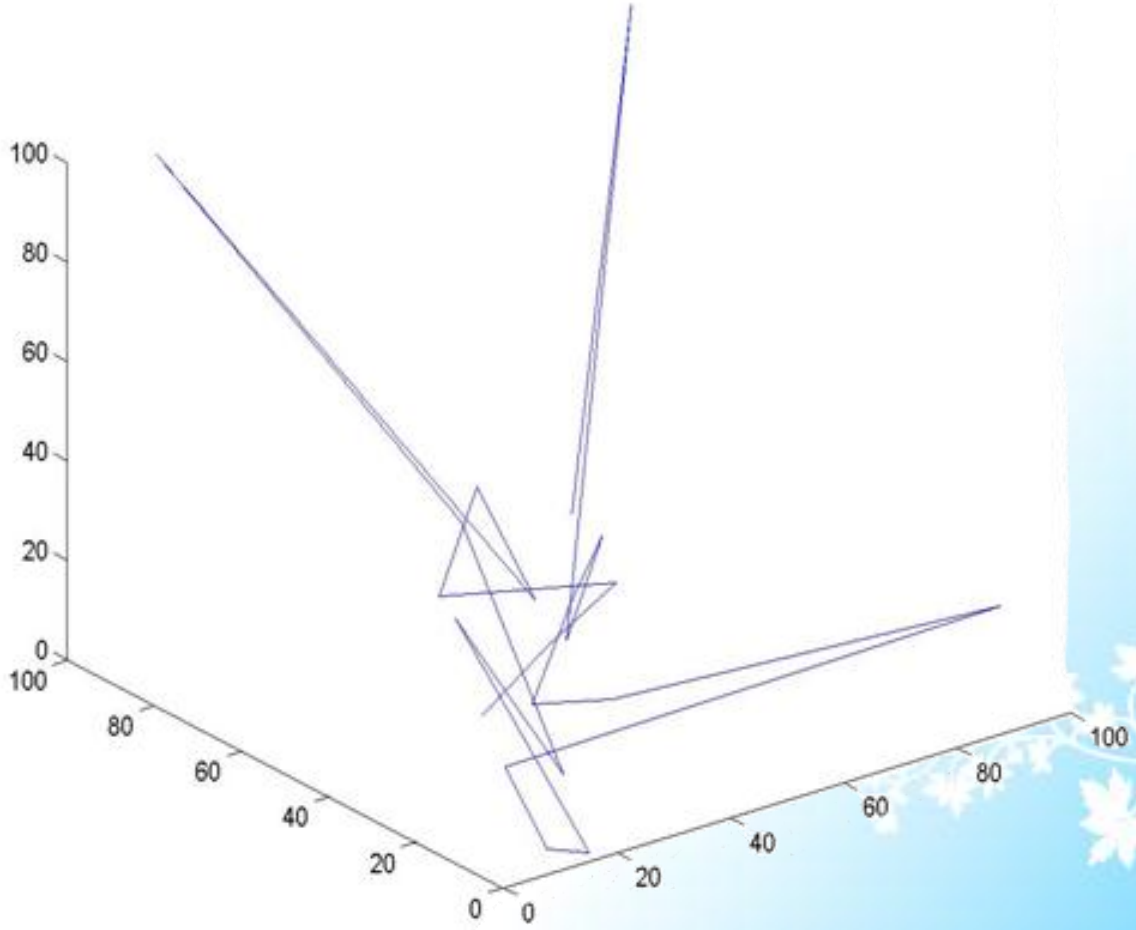


Figure: Line plot of 3-D data.

Cont..

Mesh and Surface plots:

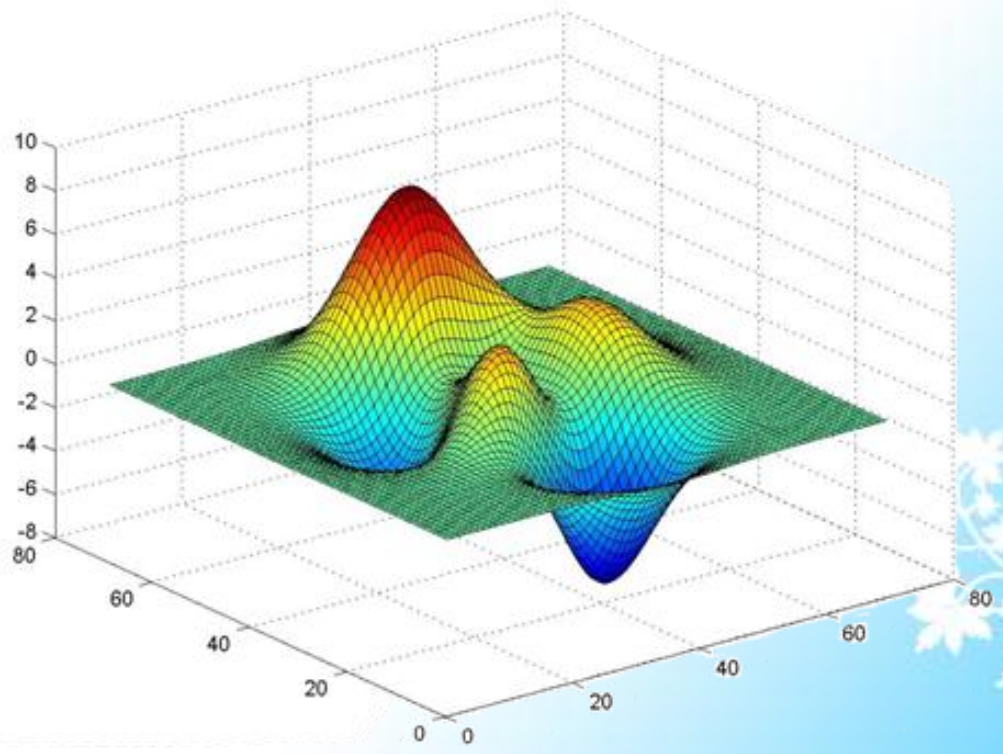
Surface information can enhance by controlling color parameters of the plot. Color parameters can map particular data values to colors specified within the range called color map. To enhance surface information indexed color and true color techniques are used. Indexed color technique assigns data points to an index into the figures color map. Whereas true color specify colors such as RGB triplets. True color requires minimum 24 bit display system. Surface plot can create by providing no explicit color data. In such cases MATLAB generates color map indices from z-data. Otherwise the z-data size array should specify color data. In another case m-by-n-by-3 array of color data defines an RGB triplets use for true color.

Cont..

Example:

>> p=peaks(70);

>> surf(p)



Cont..

Creating 3-D Patches:

Patch for graphics object is made up of one or more than one polygon that may or may not be connected. Patches are used for real world object modeling such as air planes or automobiles and for drawing 2-D or 3-D polygons of arbitrary shape. the MATLAB function is used to create patch objects fill, fill3, isosurface, isocaps, some of the contour functions and patch. Patches can specify by vertices coordinate and color data. Patches support different options of color that are useful for superimposing the data on geometric shapes visualization. Patches are of two forms: 1) high-level syntax and 2) low-level syntax. In high-level syntax MATLAB automatically determines how to color each face based on specified color data. This format is not necessary to give property names for coordinates and color data. In low level syntax only property names or property value can be specified as arguments. It does not automatically color faces until we will not change the color value or property.

Syntax:

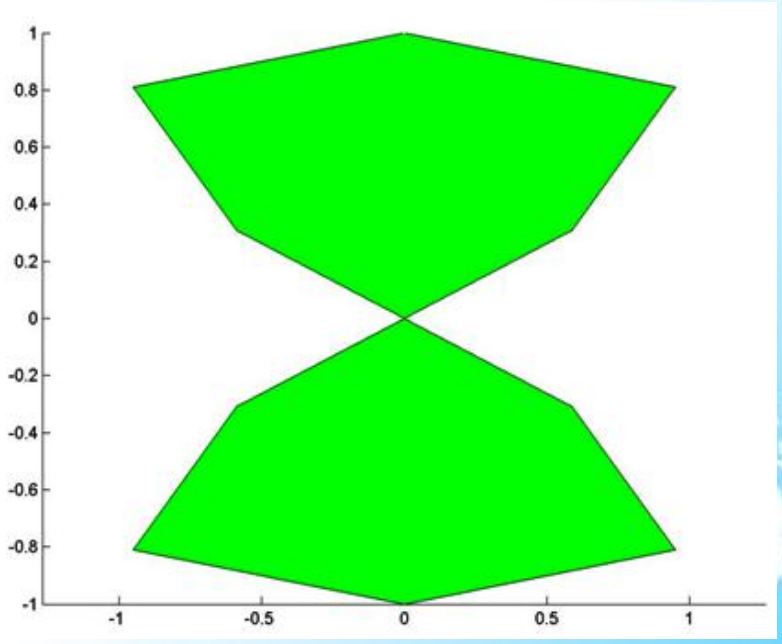
patch(x-coordinates, y-coordinates, [z-coordinates],colordata);

For more information see help patch.

Cont..

Example:

```
>> t=0:pi/5:2*pi;
>> patch(sin(2*t),cos(t),'y')
>> axis equal
```



Summary

This chapter covers the basic concept of 2-D and 3-D plotting, overview of specialized plotting, handle graphics object, visualization technique, and formatting of plotting etc. MATLAB has several high-level graphical routines. They allow a user to create various graphical objects including 2-D and 3-D graphs.

Thank You