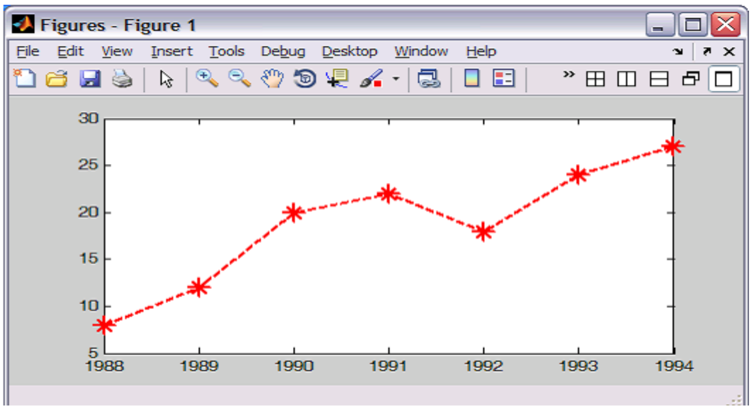
Example: plot the following table contains sales data of a company from 1988 to 1994.

Year	1988	1989	1990	1991	1992	1993	1994
Sales (millions)	8	12	20	22	18	24	27

```
>> yr=[1988:1:1994];
>> sle=[8 12 20 22 18 24 27];
>> plot(yr,sle,'--r*')
```

Line Specifiers: dashed red line and asterisk marker.



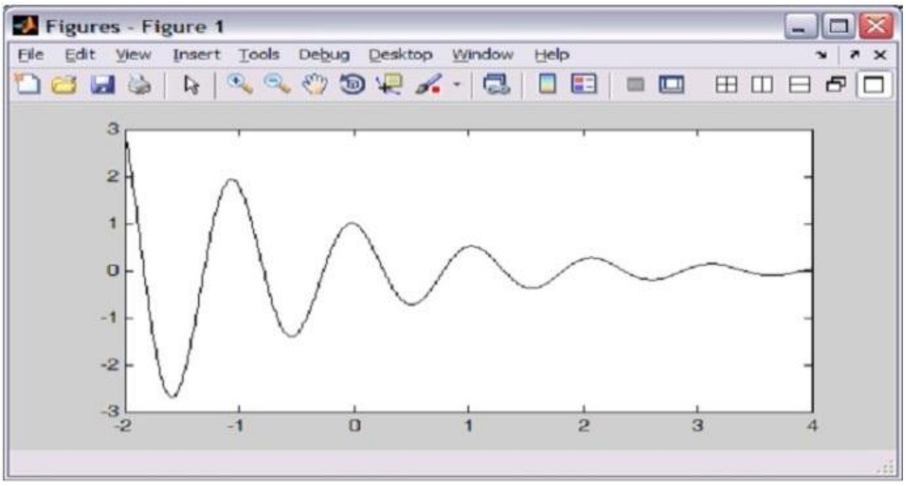
plotting a function with matlab

1. **THE *plot* COMMAND**

In order to with the plot command, the user needs to first create a vector of values of x for the domain over which the function will be plotted. Then a vector y is created with the corresponding values of by using element-by-element calculations.

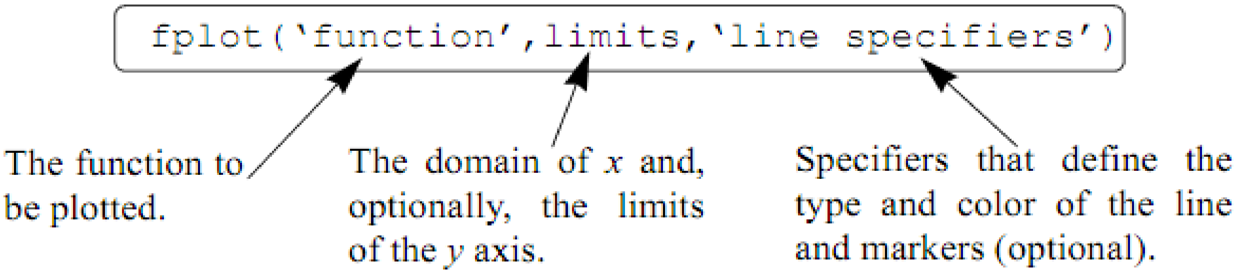
As an example, the `plot` command is used to plot the function $y = 3.5^{-0.5x}\cos(6x)$ for $-2 \le x \le 4$. A program that plots this function is shown in the following script file.

```
% A script file that creates a plot of
% the function: 3.5.^(-0.5*x).*cos(6x)
x=[-2:0.01:4];                    Create vector x with the domain of the function.
y=3.5.^(-0.5*x).*cos(6*x);        Create vector y with the function value at each x.
plot(x,y)                         Plot y as a function of x.
```



2. THE fplot COMMAND

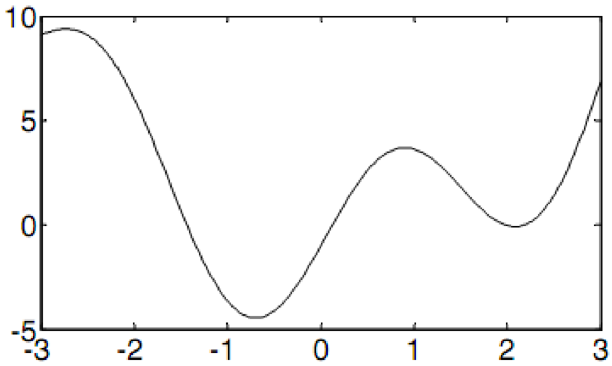
The `fplot` command plots a function with the form $y = f(x)$ between specified limits. The command has the form:



For example, a plot of the function $y = x^2 + 4\sin(2x) - 1$ for $-3 \le x \le 3$ be created with the `fplot` command by typing:

```
>> fplot('x^2+4*sin(2*x)-1',[-3 3])
```

The figure that is obtained in the Figure Window is shown in Figure



Plotting multiple graphs in the same plot

In many situations there is a need to make several graphs in the same plot. One is by using the plot command, the second is by using the hold on and hold off commands, and the third is by using the line command

1. Using the plot Command

Two or more graphs can be created in the same plot by typing pairs of vectors inside the `plot` command. The command

```
plot(x,y,u,v,t,h)
```

creates three graphs—`y` vs. `x`, `v` vs. `u`, and `h` vs. `t`—all in the same plot. The vectors of each pair must be of the same length. MATLAB automatically plots the graphs in different colors so that they can be identified. It is also possible to add line specifiers following each pair. For example the command

```
plot(x,y,'-b',u,v,'--r',t,h,'g:')
```

Example:

Plot the function $y = 3x^3 - 26x + 10$, and its first and second derivatives, for $-2 \le x \le 4$, all in the same plot.

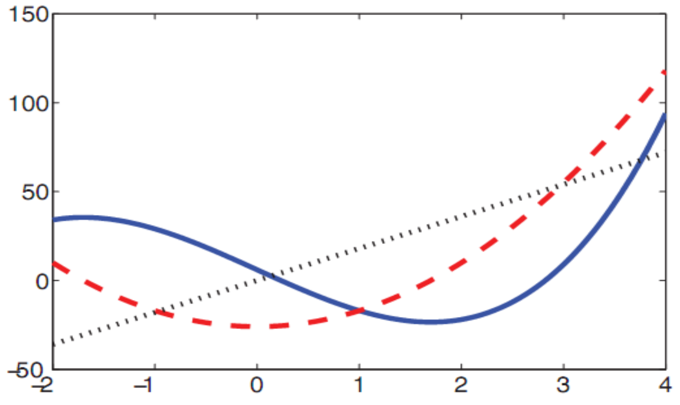
The first derivative of the function is: $y' = 9x^2 - 26$.
The second derivative of the function is: $y'' = 18x$.
A script file that creates a vector x and calculates the values of y, y', and y'' is:

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
plot(x,y,'-b',x,yd,'--r',x,ydd,':k')
```

- `x=[-2:0.01:4];` — Create vector x with the domain of the function.
- `y=3*x.^3-26*x+6;` — Create vector y with the function value at each x.
- `yd=9*x.^2-26;` — Create vector yd with values of the first derivative.
- `ydd=18*x;` — Create vector ydd with values of the second derivative.
- `plot(x,y,'-b',x,yd,'--r',x,ydd,':k')` — Create three graphs, y vs. x, yd vs. x, and ydd vs. x, in the same figure.

2. Using the *hold on* and *hold off* Commands

To plot several graphs using the hold on and hold off commands, one graph is plotted first with the plot command. Then the hold on command is typed. This keeps the Figure Window with the first plot open, including the axis properties and formatting, As an example, a solution of last Problem using the hold on and hold off commands is shown in the following script file:



```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
plot(x,y,'-b')
hold on
plot(x,yd,'--r')
plot(x,ydd,':k')
hold off
```

The first graph is created.

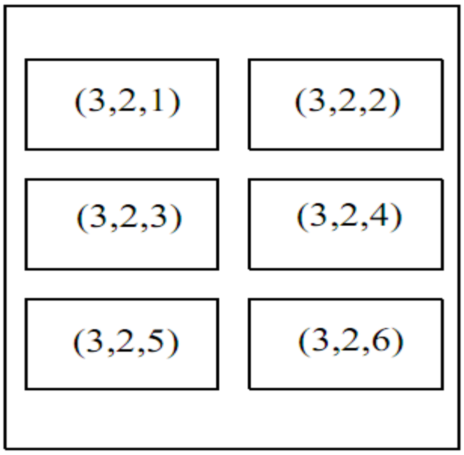
Two more graphs are added to the figure.

1. Putting multiple plots on the same page

Multiple plots can be created on the same page with the subplot command, which has the form:

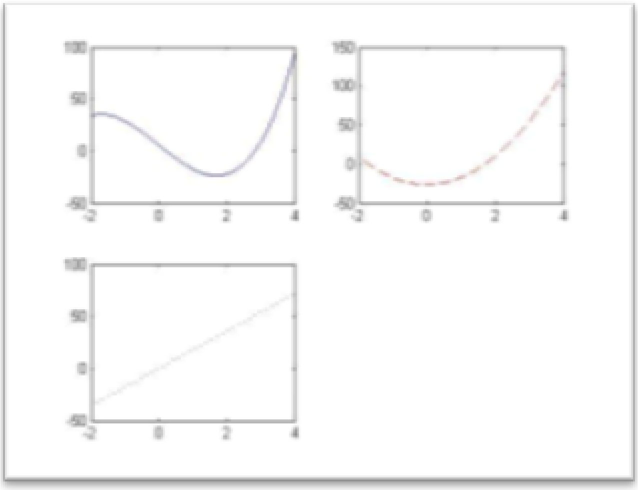
```
subplot(m,n,p)
```

The command divides the Figure Window (and the page when printed) into $m \times n$ rectangular subplots. The subplots are arranged like elements in an $m \times n$ matrix where each element is a subplot. The subplots are numbered from 1 through $m \cdot n$. The upper left subplot is numbered 1 and the lower right subplot is numbered $m \cdot n$. The numbers increase from left to right within a row, from the first row to the last. The command `subplot(m,n,p)` makes the subplot `p` current.



If we solution last example using sub plot getting:

```
clear all
clc
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
subplot(2,2,1),plot(x,y,'-b')
subplot(2,2,2),plot(x,yd,'--r')
subplot(2,2,3),plot(x,ydd,':k')
```



Formatting a Plot Using Commands:

The formatting commands are entered after the `plot` or the `fplot` command. The various formatting commands are:

The `xlabel` and `ylabel` commands:

Labels can be placed next to the axes with the `xlabel` and `ylabel` command which have the form:

```
xlabel('text as string')
ylabel('text as string')
```

The `title` command:

A title can be added to the plot with the command:

```
title('text as string')
```

The `legend` command:

The `legend` command places a legend on the plot. The legend shows a sample of the line type of each graph that is plotted, and places a label, specified by the user, beside the line sample. The form of the command is:

```
legend('string1','string2', ..... ,pos)
```

The `text` command:

A text label can be placed in the plot with the `text` or `gtext` commands:

```
text(x,y,'text as string')
gtext('text as string')
```

The `text` command places the text in the figure such that the first character is positioned at the point with the coordinates `x`, `y` (according to the axes of the figure). The `gtext` command places the text at a position specified by the user. When the command is executed, the Figure Window opens and the user specifies the position with the mouse.

The `axis` command:

When the `plot(x,y)` command is executed, MATLAB creates axes with limits that are based on the minimum and maximum values of the elements of `x` and `y`. The `axis` command can be used to change the range and the appearance of the axes. In many situations a graph looks better if the range of the axes extend beyond the range of the data. The following are some of the possible forms of the `axis` command:

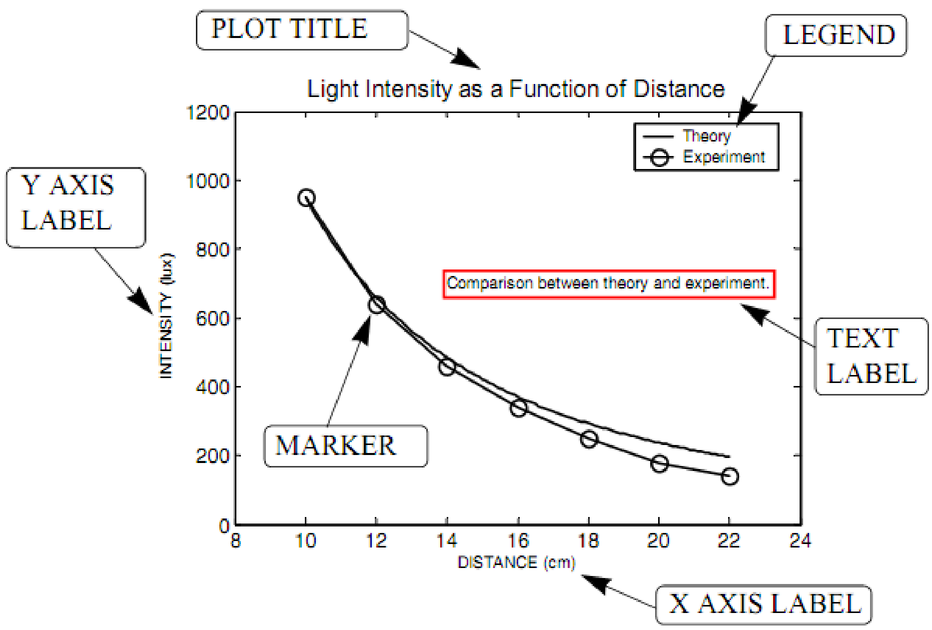
`axis([xmin,xmax,ymin,ymax])` Sets the limits of both the *x* and *y* axes (`xmin`, `xmax`, `ymin`, and `ymax` are numbers).

`axis equal` Sets the same scale for both axes.

`axis square` Sets the axes region to be square.

`axis tight` Sets the axis limits to the range of the data.

An example of formatting a plot by using commands is given in the following script file which was used to generate the formatted plot:

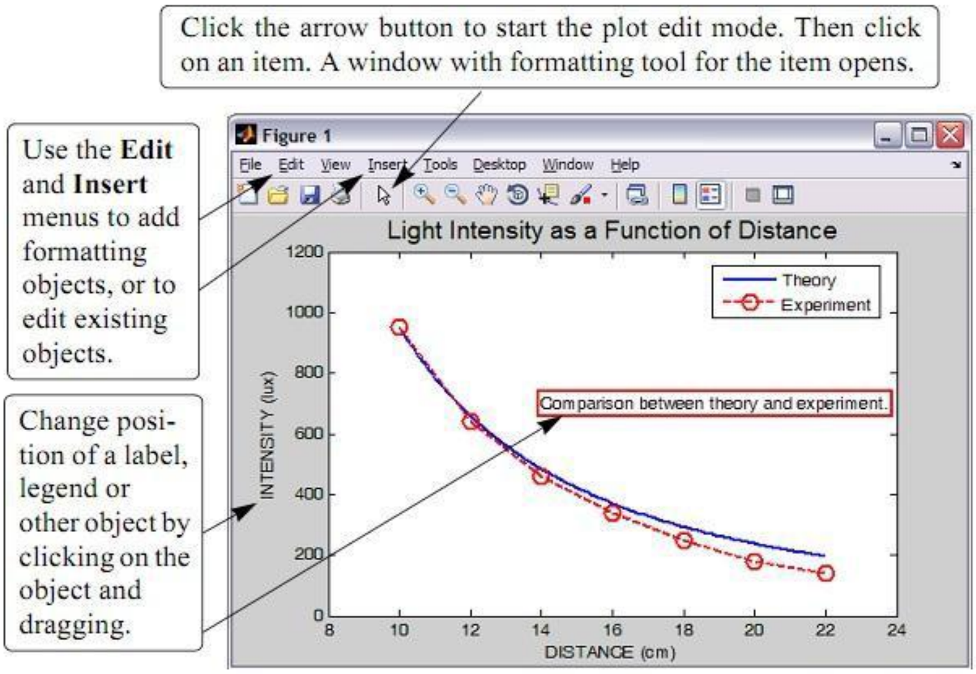


The code:

```
x=[10:0.1:22];
y=95000./x.^2;
xd=[10:2:22];
yd=[950 640 460 340 250 180 140];
plot(x,y,'-','LineWidth',1.0)
xlabel('DISTANCE (cm)')
ylabel('INTENSITY (lux)')
title('\fontname{Arial}Light Intensity as a Function of Distance','FontSize',14)
axis([8 24 0 1200])
gtext('Comparison between theory and experiment.','EdgeColor','r','LineWidth',2)
hold on
plot(xd,yd,'ro--','linewidth',1.0,'markersize',10)
legend('Theory','Experiment',0)
hold off
```

Formatting a Plot Using the Plot Editor

A plot can be formatted interactively in the Figure Window by clicking on the plot and/or using the menus. Figure 5-8 shows the Figure Window with the plot of Figure. The Plot Editor can be used to introduce new formatting items or to modify formatting that was initially introduced with the formatting commands.

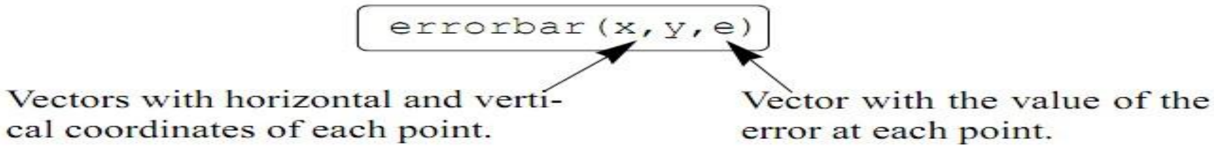


Formatting a plot using the Plot Editor.

PLOTS WITH SPECIAL GRAPHICS

1. PLOTS WITH ERROR BARS

Plots with error bars can be done in MATLAB with the error bar command. Two forms of the command, one for making plots with symmetric error, the error bar extends the same length above and below the data point and the command has the form:



The plot in Figure, which has symmetric error bars, was done by executing the following code:

xd=[10:2:22];

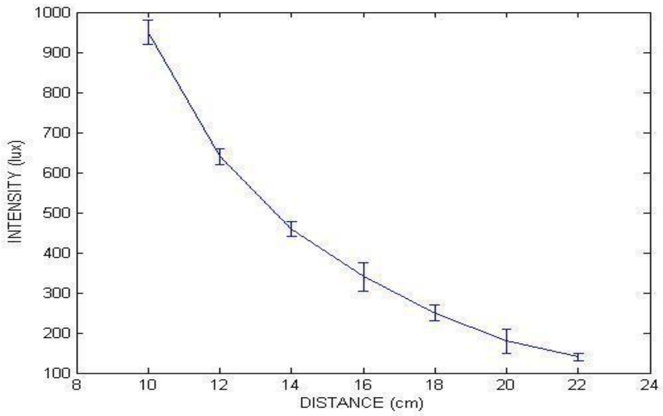
yd=[950 640 460 340 250 180 140];

ydErr=[30 20 18 35 20 30 10]

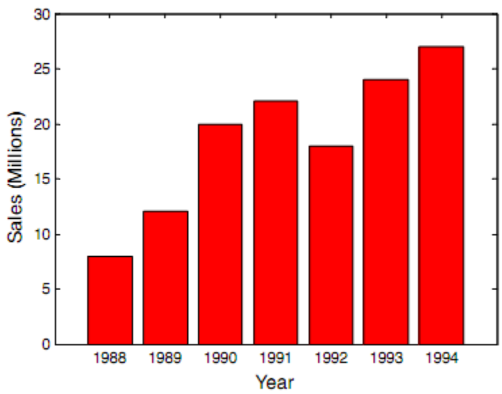
errorbar(xd,yd,ydErr)

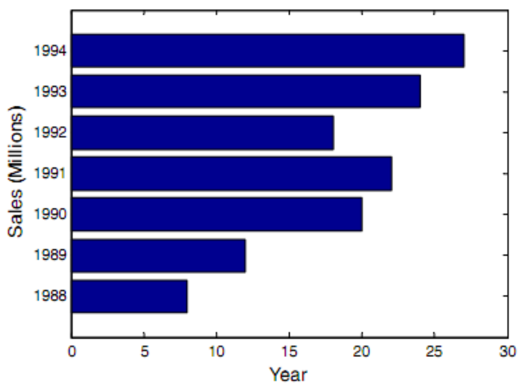
xlabel('DISTANCE (cm)')

ylabel('INTENSITY (lux)')

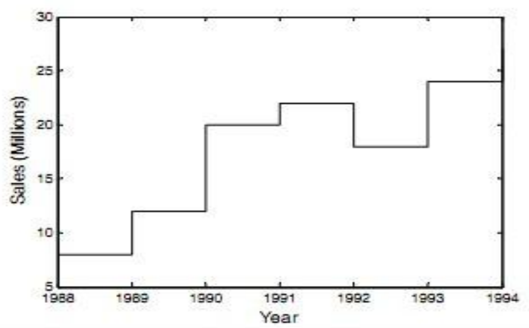


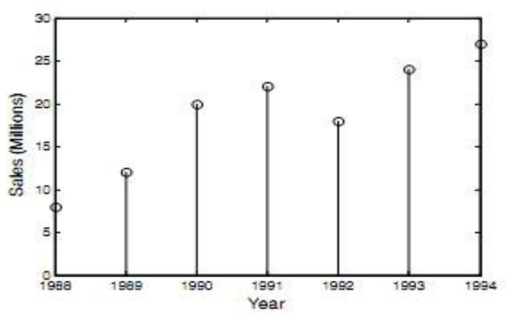
2. Bar (vertical and horizontal) plot, are presented in the following charts using the sales data**:**

Vertical Bar Plot
 	
Function format:
 	
`bar(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`bar(yr,sle,'r')` ← The bars are in red.
		`xlabel('Year')`
		`ylabel('Sales (Millions)')`

Horizontal Bar Plot
 	
Function format:
 	
`barh(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`barh(yr,sle)`
		`xlabel('Sales (Millions)')`
		`ylabel('Year')`

3. Stairs and stem plot As following

<u>Stairs Plot</u>
 	
Function format:
 	
`stairs(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`stairs(yr,sle)`

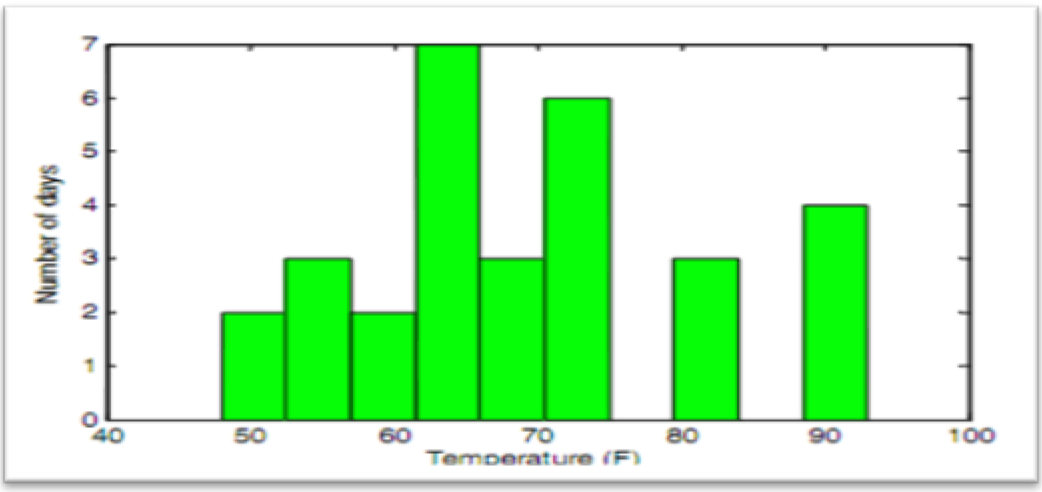
<u>Stem Plot</u>
 	
Function Format
 	
`stem(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`stem(yr,sle)`

4. Histogram plot Histograms are created in MATLAB with the hist command. The simplest form of the command is:

```
hist(y)
```

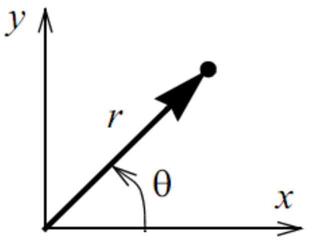
y is a vector with the data points. MATLAB divides the range of the data points into 10 equally spaced sub ranges (bins) and then plots the number of data points in each bin. For example, the following data points are the daily maximum temperature (in F) in Washington, DC, during the month of April 2002: 58 73 73 53 50 48 5673 73 66 69 63 74 82 84 91 93 89 91 80 59 69 56 64 63 66 64 74 63 69 (data from the U.S. National Oceanic and Atmospheric Administration). A histogram of this data is obtained with the commands:

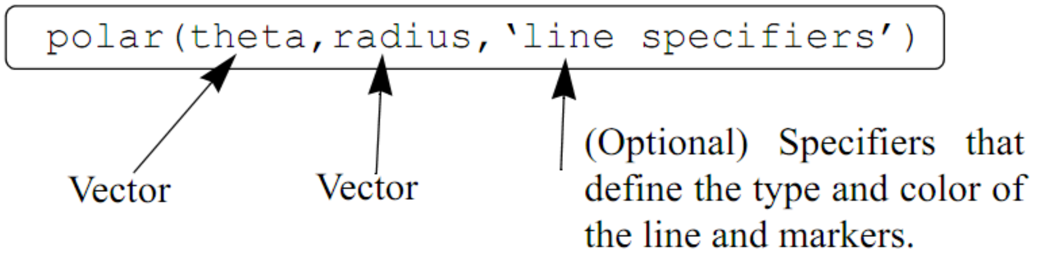
```
>> y=[58 73 73 53 50 48 56 73 73 66 69 63 74 82 84 91 93 89
91 80 59 69 56 64 63 66 64 74 63 69];
>> hist(y)
```



5. POLAR PLOTS

Polar coordinates, in which the position of a point in a plane is defined by the angle θ and the radius (distance) to the point, are frequently used in the solution of science and engineering problems. The `polar` command is used to plot functions in polar coordinates. The command has the form:

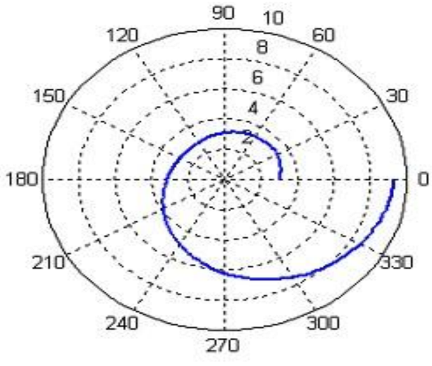




element calculations. The two vectors are then used in the `polar` command.

For example, a plot of the function $r = 3\cos^2(0.5\theta) + \theta$ for $0 \le \theta \le 2\pi$ is shown below.

```
t=linspace(0,2*pi,200);
r=3*cos(0.5*t).^2+t;
polar(t,r)
```



Three-Dimensional Plots

1. LINE PLOTS

A three-dimensional line plot is a line that is obtained by connecting points in three-dimensional space. A basic 3-D plot 'is created with the `plot3` command, which is very similar to the `plot` command and has the form:

```
plot3(x,y,z,'line specifiers','PropertyName',property value)
```

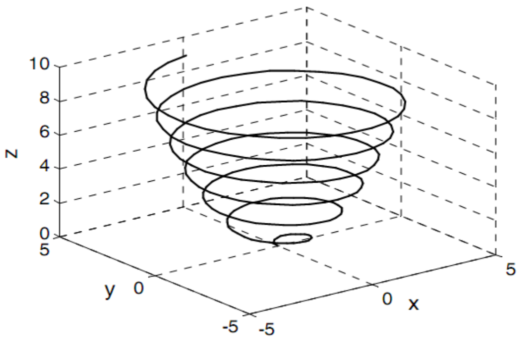
- x, y, and z are vectors of the coordinates of the points.
- (Optional) Specifiers that define the type and color of the line and markers.
- (Optional) Properties with values that can be used to specify the line width, and marker's size and edge and fill colors.

For example, if the coordinates x, y, and z are given as a function of the parameter t by

$$x = \sqrt{t}\sin(2t)$$
$$y = \sqrt{t}\cos(2t)$$
$$z = 0.5t$$

a plot of the points for $0 \le t \le 6\pi$ can be produced by the following script file:

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
plot3(x,y,z,'k','linewidth',1)
grid on
xlabel('x'); ylabel('y'); zlabel('z')
```



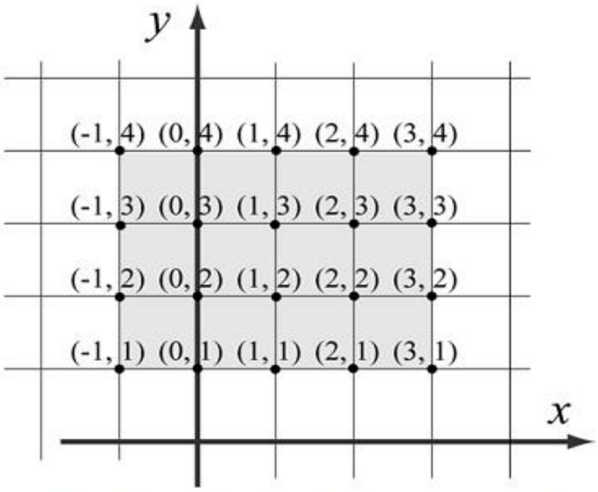
MESH AND SURFACE PLOTS

Mesh and surface plots are three-dimensional plots used for plotting functions of the form z =f(x,y) where *x* and *y* are the independent variables and *z* is the dependent variable. It means that within a given domain the value of *z* can be calculated for any combination of *x* and *y*. Mesh and surface plots are created in three steps.

The first step is to create a grid in the x y plane that covers the domain of the function. The second step is to calculate the value of z at each point of the grid. The third step is to create the plot.

Creating a grid in the *x y* plane (Cartesian coordinates):

The grid is a set of points in the x y plane in the domain of the function. The density of the grid (number of points used to define the domain) is defined by the user. Figure 10-2 shows a grid in the domain $-1 \le x \le 3$ and $1 \le y \le 4$. In this grid



A grid in the x y plane for the domain $-1 \le x \le 3$ and $1 \le y \le 4$ with spacing of 1.

the distance between the points is one unit. The points of the grid can be defined by two matrices, X and Y. Matrix X has the x coordinates of all the points, and matrix Y has the y coordinates of all the points:

$$X = \begin{bmatrix} -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

MATLAB has a built-in function, called **meshgrid**, that can be used for creating the X and Y matrices. The form of the **meshgrid** function is:

```
[X,Y] = meshgrid(x,y)
```

X is the matrix of the x coordinates of the grid points.
Y is the matrix of the y coordinates of the grid points.

x is a vector that divides the domain of x.
y is a vector that divides the domain of y.

```
x=-3:3
y=-3:1
 [X,Y]=meshgrid(x,y)
```

Ans:

X =

-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3

Y =

-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0
1	1	1	1	1	1	1

Making mesh and surface plots:

A mesh or surface plot is created with the `mesh` or `surf` command, which has the form:

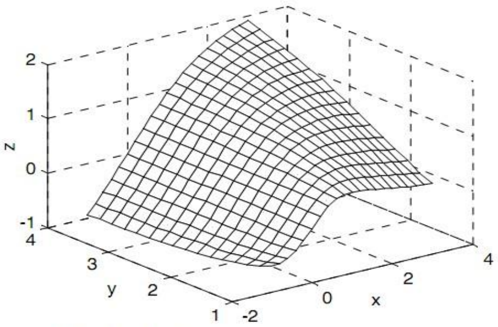
`mesh(X,Y,Z)`   `surf(X,Y,Z)`

where `X` and `Y` are matrices with the coordinates of the grid and `Z` is a matrix with the value of z at the grid points. The mesh plot is made of lines that connect the points. In the surface plot, areas within the mesh lines are colored.

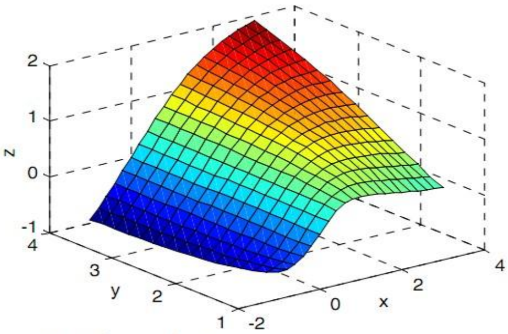
As an example, the following script file contains a complete program that creates the grid and then makes a mesh (or surface) plot of the function $z = \frac{xy^2}{x^2+y^2}$ over the domain $-1 \le x \le 3$ and $1 \le y \le 4$.

```
x=-1:0.1:3;
y=1:0.1:4;
[X,Y]=meshgrid(x,y);
Z=X.*Y.^2./(X.^2+Y.^2);
mesh(X,Y,Z)                Type surf(X,Y,Z) for surface plot.
xlabel('x'); ylabel('y'); zlabel('z')
```

Note that in the program above the vectors x and y have a much smaller spacing than the spacing earlier in the section. The smaller spacing creates a denser grid. The figures created by the program are:



Mesh plot



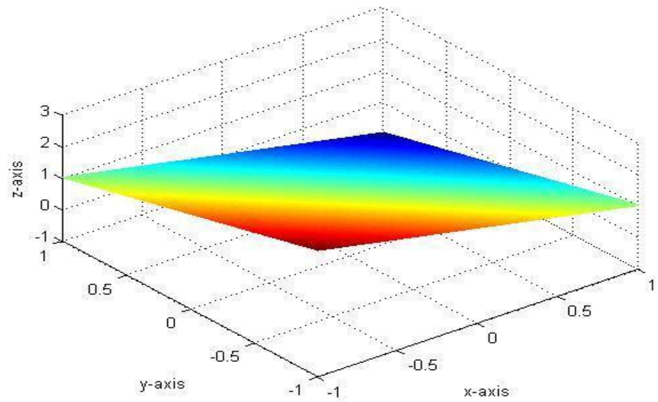
Surface plot

Exampls:

1. plot x+y+z=1 in 3D?

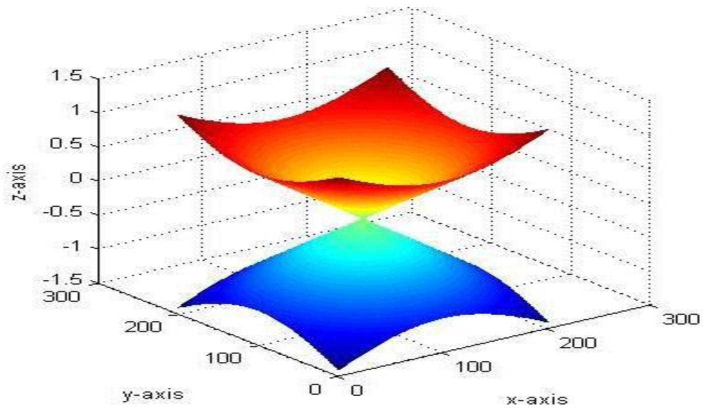
Solution:

```
x=-1:.01:1;
y=x;
[x y]=meshgrid(x,y);
z=1-x-y;
mesh(x,y,z)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```



2. plot $x^2+y^2=z^2$ in 3D?

```
x=-1:.01:1;
y=x;
[x y]=meshgrid(x,y);
z1=sqrt(x.^2+y.^2); z2=-
sqrt(x.^2+y.^2);
      mesh(z1)
       hold on
      mesh(z2)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```



PROBLEMS

Use the `fplot` command to plot the function
$f(x) = \sqrt{|\cos(3x)|} + \sin^2(4x)$ in the domain $-2 \le x \le 2$.

Make two separate plots of the function $f(x) = (x+1)(x-2)(2x-0.25) - e^x$, one plot for $0 \le x \le 3$ and one for $-3 \le x \le 6$.

Plot $x^2+y^2=z^2+1$ (Hyperbloid of one sheet) in 3D?

plot $x^2-y^2=z$ (Elliptic paraboloid) in 3D.

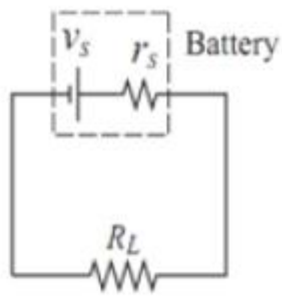
Prove (by plot) the equation ($x^2+y^2=16$) is the circle in 2D , and cylindrical in 3D?

plot the data x=(3 4 6 8 10), y=(12 14 16 20 22) ,using Bars,Stairs and stem plots?

An electrical circuit that includes a voltage source v_S with an internal resistance r_S and a load resistance R_L is shown in the figure. The power P dissipated in the load is given by

$$P = \frac{v_S^2 R_L}{(R_L + r_S)^2}$$

Plot the power P as a function of R_L for $1 \le R_L \le 10\,\Omega$, given that $v_S = 12$ V and $r_S = 2.5\,\Omega$.



The position of a moving particle as a function of time is given by:

$$x = (4-0.1t)\sin(0.8t) \qquad y = (4-0.1t)\cos(0.8t) \qquad z = 0.4t^{(3/2)}$$

Plot the position of the particle for $0 \le t \le 30$.