The 2D-Plot Command

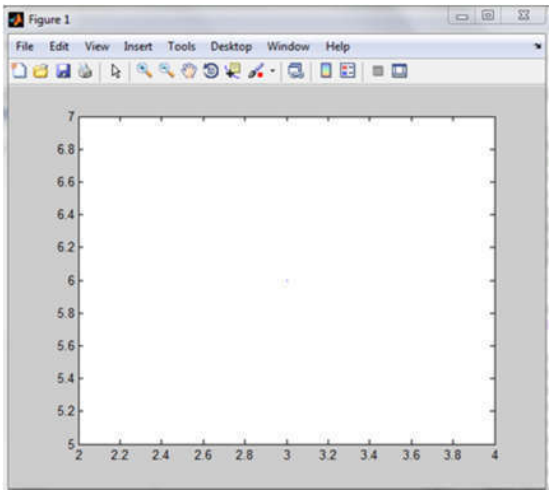
The basic plotting command in MATLAB is plot. It is take two arrangements x and y.

Plot (x,y)

for example if you want to plot a point in with x value equal to 3 and y value equal to 6 we must write the following command

>> plot(3,6)

This commend is produce the following figure



We can use variable for x and y values

```
>> x=3;
>> y=6;
>> plot(x,y)
```

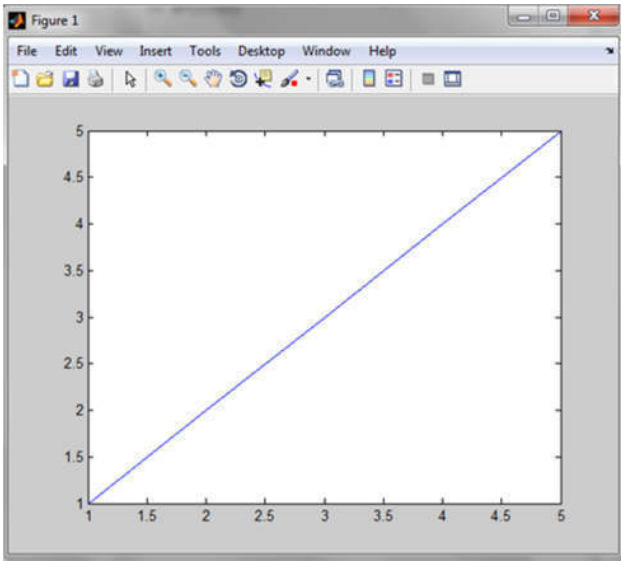
and we got the same result

If x and y are a vectors, plot (x,y) produces a graph of y versus x.

For example

```
>> x=[1:5]
x =
    1    2    3    4    5
>> y=[1:5]
y =
    1    2    3    4    5
>> plot(x,y)
```

Then MATLAB plot the following figure



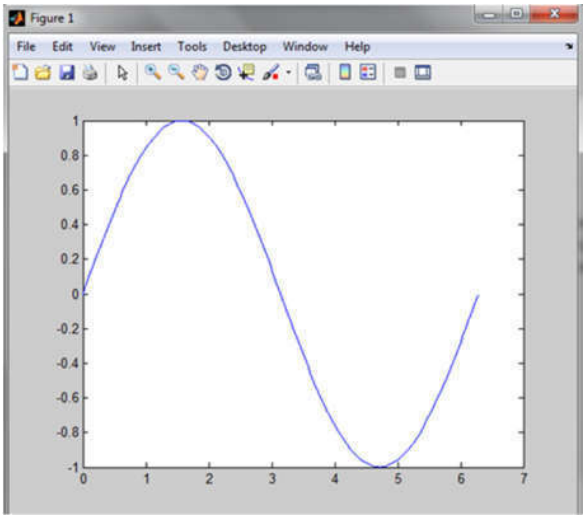
Example:

```
>> t = 0:pi/100:2*pi;
>> y = sin(t);
>> plot(t,y)
```

Note: 0:pi/100:2*pi yields a vector that

- starts at 0,
- takes steps (or increments) of pi/100
- stops when 2*pi is reached.

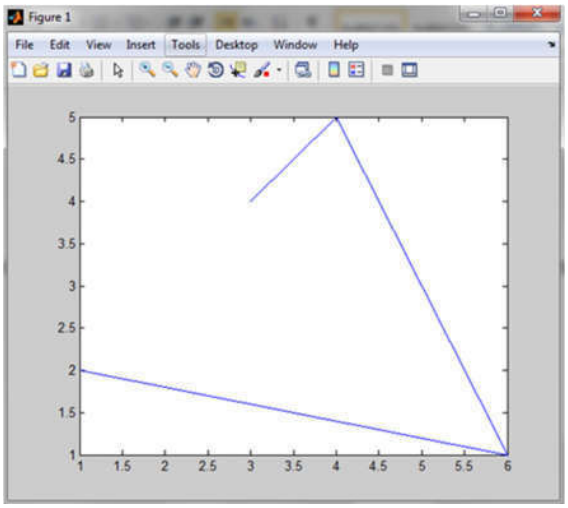
These statements create a vector of values in the range [0, 2π] in increments of π/100 and then use this vector to evaluate the sine function over that range. MATLAB plots the vector on the x-axis and the value of the sine function on the y-axis. Then MATLAB plot the following figure.



Example:

```
>> x=[3,4,6,1]
x =
   3    4    6    1

>> y=[4,5,1,2]
y =
   4    5    1    2

>> plot(x,y)
```

The result will be the figure bellow.



Note:

To use the 'plot' function in MATLAB, you should first make sure that the /vectors you are trying to use are of equal dimensions.

Example:

If we change the size of x vector in pervious example

```
>> x=[3,4,6]
x =
   3    4    6
>> y=[4,5,1,2]
y =
   4    5    1    2
>>  plot(x,y)
??? Error using ==> plot
```

Vectors must be the same lengths.

As we see error message appear because vector x size is 3 and it not equal to the size of vector y (size of y vector is 4).

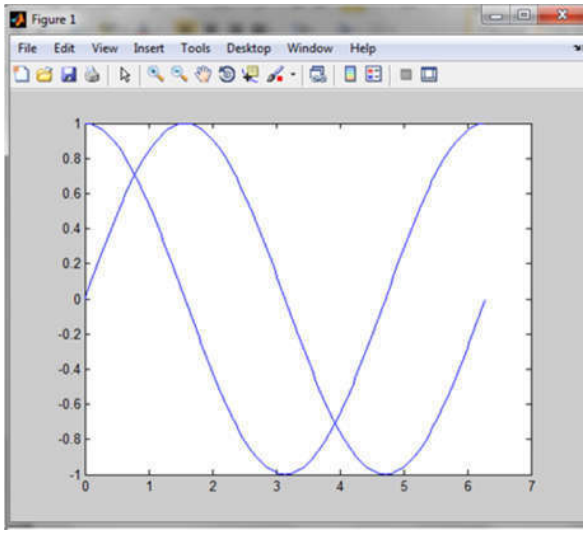
Plot Multiple Lines on the same Axes

Crating a plot with more than one line can be accomplished in several ways. By default, the execution of the second plot statement will erase the first plot. However, you can layer plots on top of one another by using **hold on** command.

Example :

```
>> x=0:pi/100:2*pi;
>> y1=cos(x);
>> plot(x,y1)
>> y2=sin(x);
>> hold on;
>> plot(x,y2)
```

By executing this statements we create a plot with both functions plotted on the same graph as shown in the figure below



Semicolons are optional on both plot and the hold on statement.

MATLAB will continue to layer the plots until the **hold off** command is executed:

```
>> hold off
```

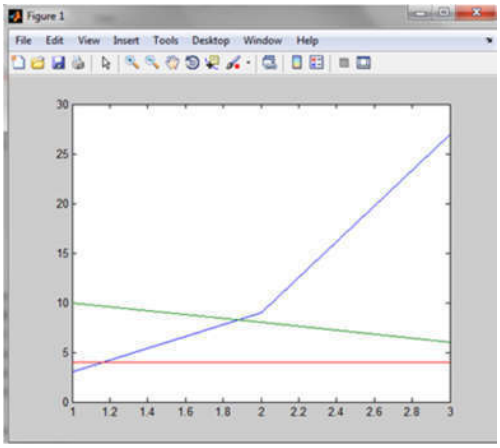
Another way to create a graph with multiple lines is to request both lines in single plot command as alternating x and y vectors as in

Plot(x,y,w,z)

Example:

```
>>  x=[3 9 27];
>> y=[10 8 6];
>>  z=[4 4 4];
>> t=[1 2 3];
>> plot(t, x, t, y, t, z)
```

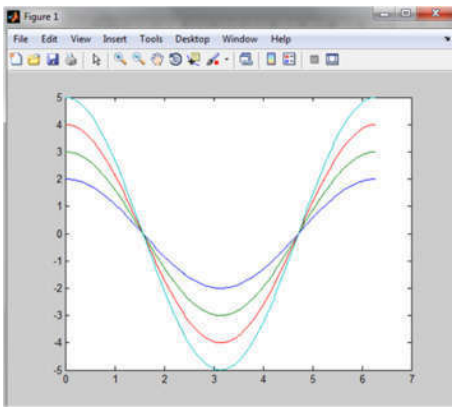
You can plot multiple graphs in one call to plot using x-y pairs. MATLAB automatically cycles through a predefined list of colors to allow discrimination between each set of data.



Example

```
>> x=0:pi/100:2*pi;
>> y1=cos(x)*2;
>> y2=cos(x)*3;
>> y3=cos(x)*4;
>> y4=cos(x)*5;
>> z=[y1;y2;y3;y4];
>> plot(x,y1,x,y2,x,y3,x,y4);
```

This produces the result shown in figure below.



The same result is obtain with Plot(x,z);

```
>> x=0:pi/100:2*pi;
>> y1=cos(x)*2;
>> y2=cos(x)*3;
>> y3=cos(x)*4;
>> y4=cos(x)*5;
>> z=[y1;y2;y3;y4];
>> plot(x,z);
```

Line ,Color, and Mark Style

It is possible to specify line styles, colors, and markers (e.g., circles, plus signs, . . .) using the plot command:

plot(x,y,'style_color_marker')
or

plot(x1, y1, 'line style', x2, y2, 'line style'...)

- A line style (e.g., dashed, dotted, etc.)
- A marker type (e.g., x, *, o, etc.)
- A predefined color specifier (c, m, y, k, r, g, b, w)
  See table 1 for a complete list :

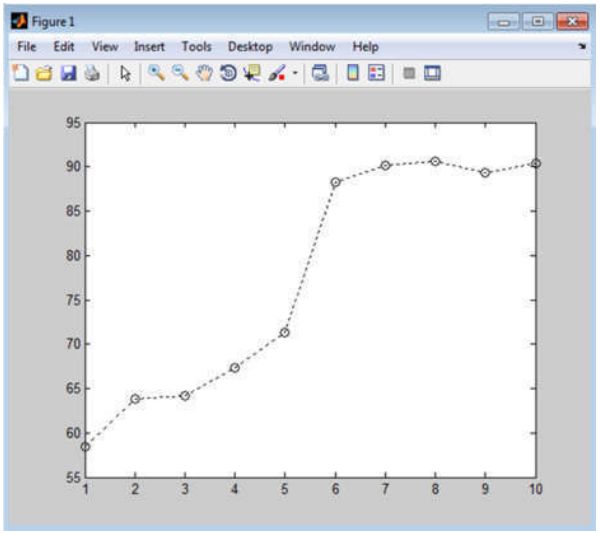
Color		markers		Line types	
Indicator	color	Indicator	Point type	Indicator	Line types
b	blue	o	circle	--	dashed
c	cyan	d	diamond	-.	dash dot
g	green	h	hexagram	:	dotted
k	black	p	pentagram	-	solid
m	magenta	+	plus		
r	red	.	point		
y	yellow	s	square		
		*	star		
		v	down triangle		
		<	left triangle		
		>	right triangle		
		^	up triangle		
		x	x-mark		

Table 1 Line type, Markers, Color.

Example:

```
>> x=[1:10];
>> y=[58.5 63.8 64.2 67.3 71.3 88.3 90.1 90.6 89.3 90.4];
>>  plot(x,y,':')
>> plot(x,y,':ok')
```

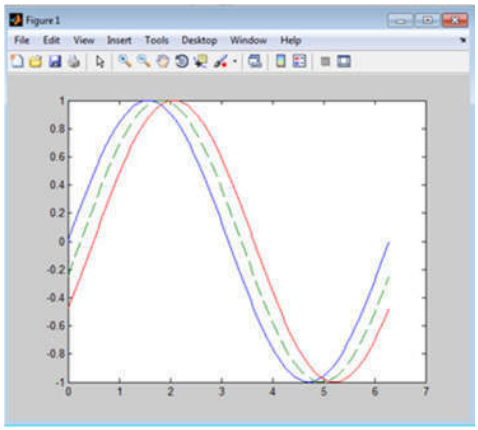
The result will be



Example:

```
>> t = 0:pi/100:2*pi;
>> y = sin(t);
>> y2 = sin(t-0.25);
>> y3 = sin(t-0.5);
>> plot(t,y,'–',t,y2,'--',t,y3,':')
```

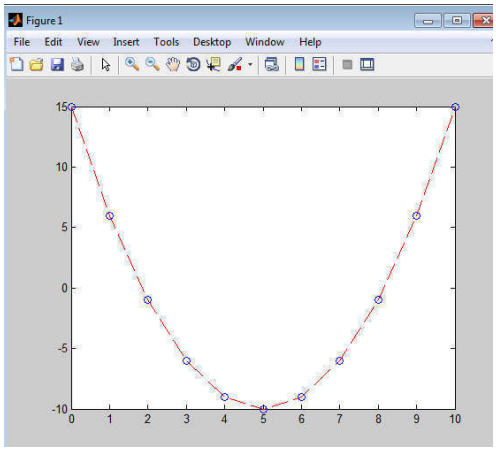
The result will be:



Example:

```
>> x=0:1:10;
>> y=x.^2-10.*x+15;
>> plot(x,y,'r--',x,y,'bo')
```

The result will be:



Plotting Only the Data Points

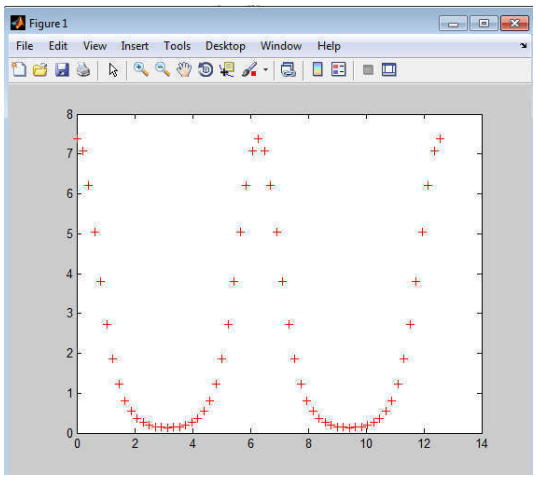
To plot a marker at each data point without connecting the markers with lines, use a specification that does not contain a line style. For example, given two vectors, for example

```
>> x = 0:pi/15:4*pi;
>> y = exp(2*cos(x));
```

Calling plot with only a color and marker specify

```
>> plot(x,y,'r+')
```

That command plots a red plus sign at each data point, and we get the flowing figure

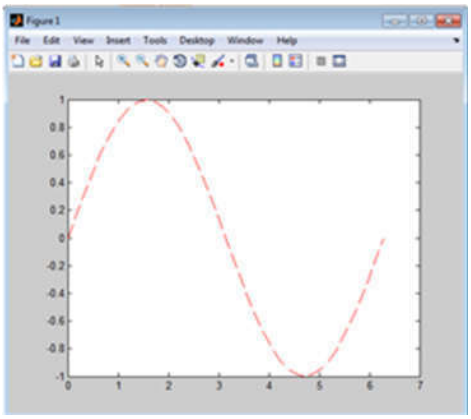


Line Width and Marker Size

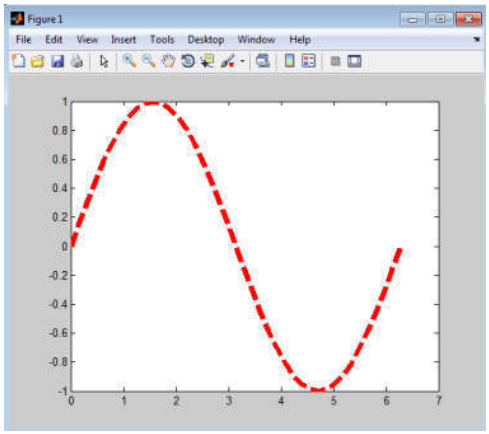
LineWidth – specifies the width of the line in units of points.

plot(x,y,' LineWidth',number)

```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

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

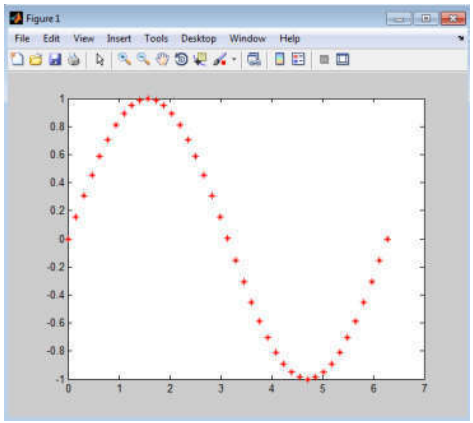


```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

>>  plot( x,y,'r--','LineWidth',5)
```

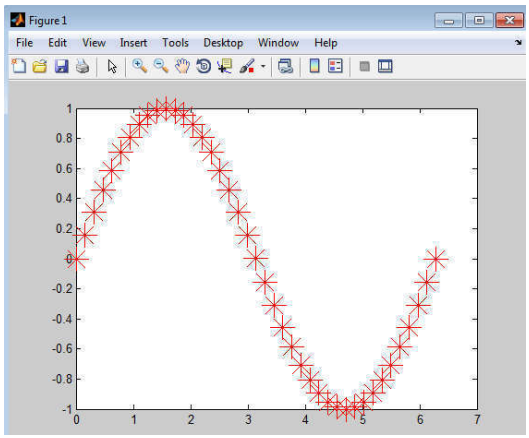


example:

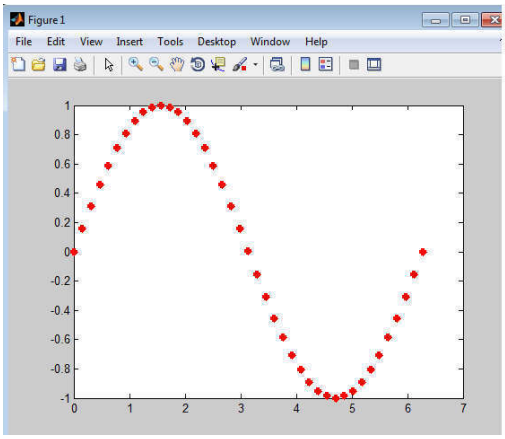
```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

>> plot( x,y,'r*')
```



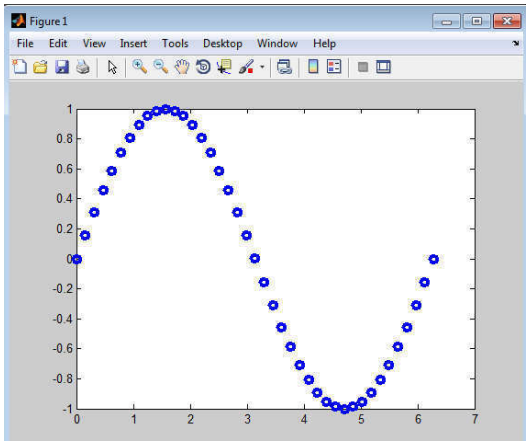
```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

>>  plot( x,y,'r*','MarkerSize',20)
```



```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

>> plot(x,y,'r*','LineWidth',3)
```




```
>> x =0: 2*pi/40: 2*pi;

>> y = sin(x);

>>  plot(x,y,'bo','LineWidth',3)
```

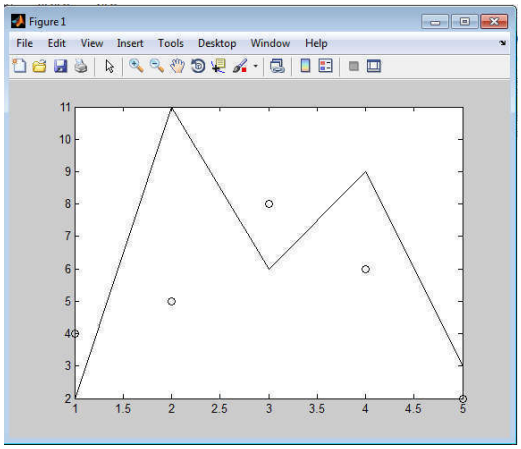


Grid line

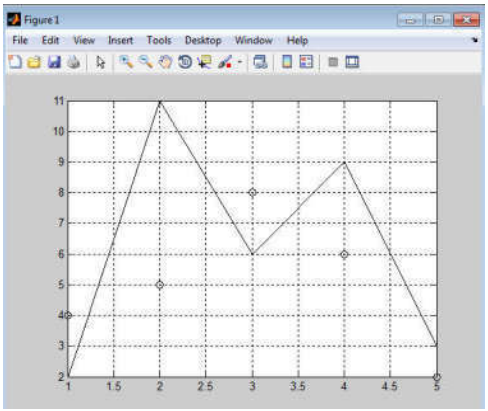
Grid displays grid lines on a graph. Called by itself, it is a toggle that turns the grid lines on and off. Alternatively, the commands grid on and grid off can be used.

Example:

```
>> x = 1:5;

>> y1 = [2 11 6 9 3];

>> y2 = [4 5 8 6 2];

>> plot( x,y1,'k')

>> hold on

>> plot( x,y2,'ko')
```




```
>> x = 1:5;

>> y1 = [2 11 6 9 3];

>> y2 = [4 5 8 6 2];

>> plot( x,y1,'k')

>> hold on

>> plot( x,y2,'ko')

>> grid on
```

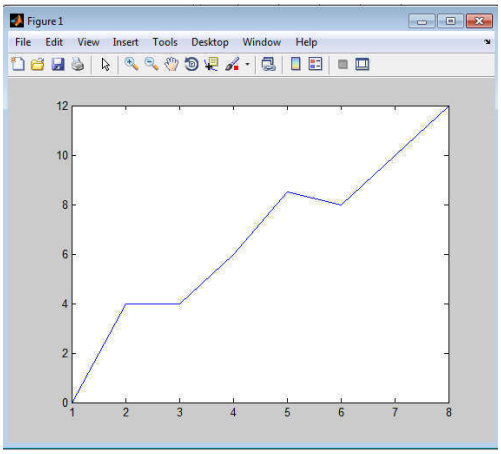


Plot with one argument

It is take the values of argument versus its index .

Example:

```
>> y=[0 4 4 6 8.5 8 10 12];

>> plot(y)
```



10-1-2017

Adding Titles to Graphs

In MATLAB, a title is a text string at the top of axes. Titles typically define the subject of the graph.

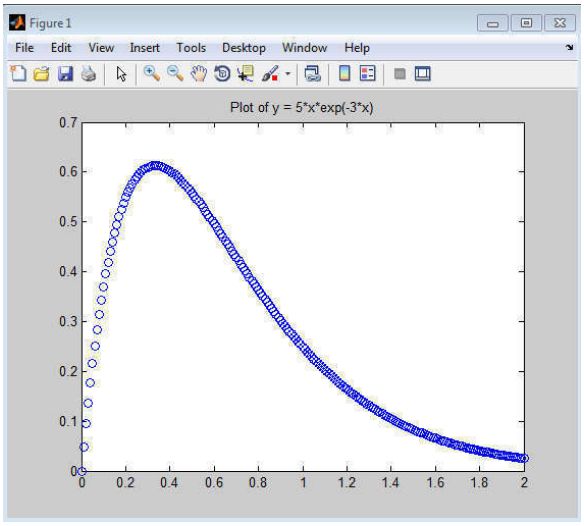
We can add title to the graph by using **title** command .

title('title –string')

this command adds the title-string as the title to the graph.

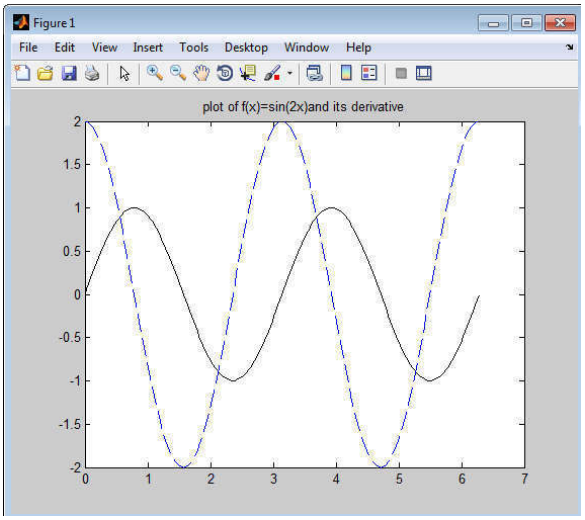
Example1:

```
>> x = 0:0.01:2;

>> y = 5*x.*exp(-3*x) ;

>> plot(x,y,'o');

>> title('Plot of y = 5*x*exp(-3*x)');

>>
```



Example2:

```
>> x=0:pi/100:2*pi;

>> y1=sin(2*x);

>> y2=2*cos(2*x);

>> plot(x,y1,'k-',x,y2,'b--');

>> title('plot of f(x)=sin(2x)and its derivative');
```



Create a multiline title

Example:

```
>> plot((1:10).^2);

>> title({'First line';'Second line'})
```