Chapter 3: Graphs

Plotting is a very important and powerful feature in MATLAB. In this chapter we will learn the basic plotting functionality in MATLAB.

Plots functions: Here are some useful functions for creating plots:

Function	Description	Example

plot	Generates a plot. plot(y) plots the columns of y against the indexes of the columns.	`>X = [0:0.01:1];`
		`>Y = X.*X;`
		`>plot(X, Y)`

figure	Create a new figure window	`>>figure` `>>figure(1)`
subplot	Create subplots in a Figure. subplot(m,n,p) or subplot(mnp), breaks the Figure window into an m-by-n matrix of small axes, selects the p-th axes for the current plot. The axes are counted along the top row of the Figure window, then the second row, etc.	`>>subplot(2,2,1)`

grid	Creates grid lines in a plot.
	"grid on" adds major grid lines to the current plot.
	"grid off" removes major and minor grid lines from the current plot.
		`>>grid`
		`>>grid on`
		`>>grid off`

axis	Control axis scaling and appearance. "axis([xmin xmax ymin ymax])" sets the limits for the x- and y-axis of the current axes.	`>>axis([xmin xmax ymin ymax])`
		`>>axis off`
		`>>axis on`

title	Add title to current plot title('string')	`>>title('this is a title')`
xlabel	Add xlabel to current plot xlabel('string')	`>> xlabel('time')`
ylabel	Add ylabel to current plot ylabel('string')	`>> ylabel('temperature')`
legend	Creates a legend in the corner (or at a specified position) of the plot	`>> legend('temperature')`
hold	Freezes the current plot, so that additional plots can be overlaid	`>>hold on` `>>hold off`

Type "**help graphics**" in the Command Window for more information, or type "**help <functionname>**" for help about a specific function.

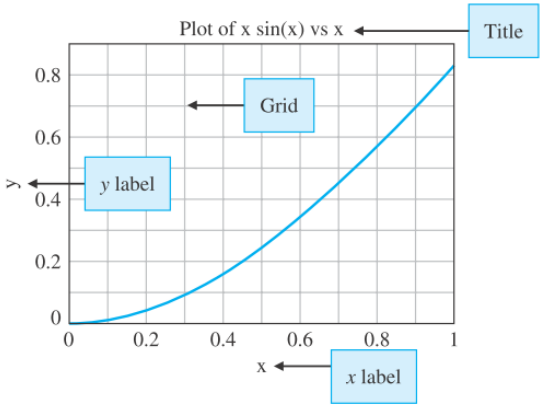
Before you start, you should use the Help system in MATLAB to read more about these functions. Type "help <functionname>" in the Command window.

Example:

Here we see some examples of how to use the different plot functions:

```
>>x=[0:0.1:1]';
>>y=x.*sin(x);
>>plot(x,y)
>>title('Plot of x sin(x) vs x ')
>>xlabel('x')
>>ylabel('y')
>>grid on
```

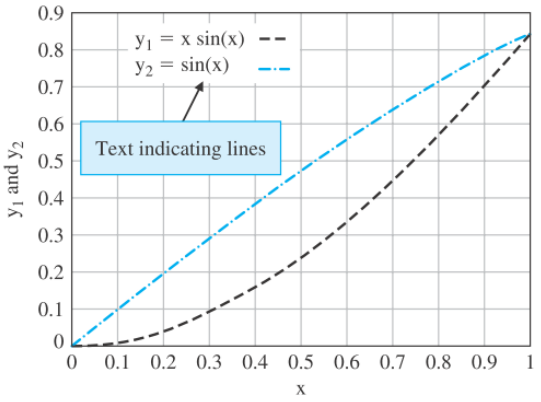
(a)




```
>>  x=[0:0.1:1]';
>>  y1=x.*sin(x); y2=sin(x);
>> plot(x,y1,'--',x,y2,'-.')
>> text(0.1,0.85,'y_1 = x sin(x) ---')
>> text(0.1,0.80,'y_2 = sin(x) .\_.\_')
>> xlabel('x'), ylabel('y_1 and y_2'), grid on
```

Dashed line for y1
Dashed-dot line for y2

(a)



[End of Example]

Task 9: Plotting

In the Command window (or use the Script Editor) in MATLAB window input the time from $t = 0$ seconds to $t = 10$ seconds in increments of 0.1 seconds as follows:

```
>>t = [0:0.1:10];
```

Then, compute the output y as follows:

```
>>y = cos(t);
```

Use the Plot command:

```
>>plot(t,y)
```

[End of Task]

Example: Find solution for 2.order equation by plotting

Assume the following algebraic equation:

$$x^2 + 4x + 3 = 0$$

We can find the solution x by using the following general term:

Given the second order algebraic equation:

$$ax^2 + bx + c = 0$$

The solution (roots) is as follows ($a \neq 0$):

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Then we can create a function that finds the solution for x based on different input values for a, b and c, e.g.,

```
function x = solveeq(a,b,c)
    x(1,1) = (-b+sqrt(b^2-4*a*c))/(2*a);
    x(2,1) = (-b-sqrt(b^2-4*a*c))/(2*a);
```

We define values for a, b and c:

```
>> a = 1;
>> b = 4;
>> c = 3,
```

Then we can use the function like this:

```
>> x = solveeq(a,b,c)
```

Then MATLAB responds with the following results:

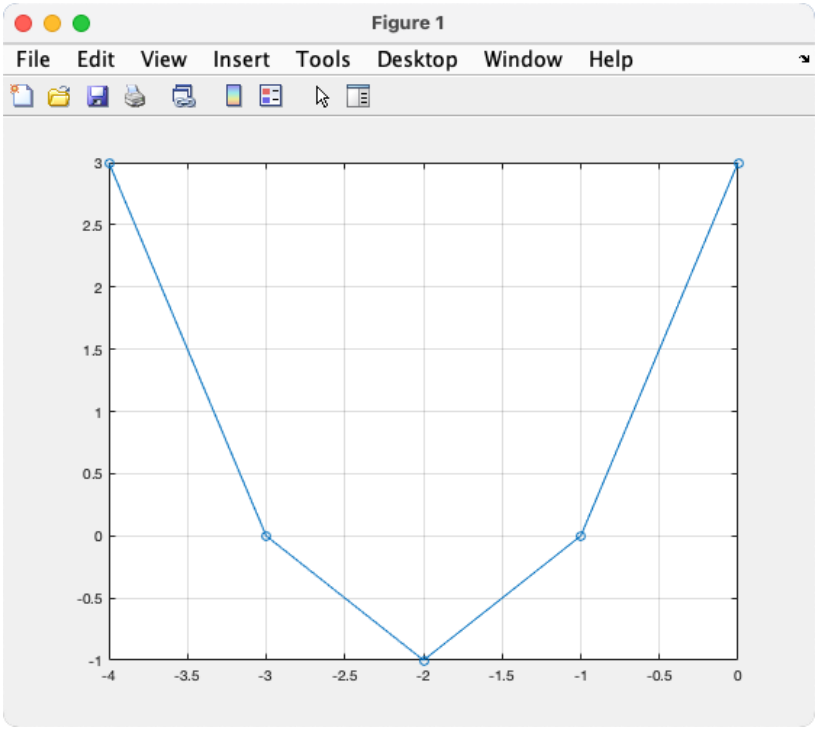
```
x =
    -1
    -3
```

But we can also make a plot and then find the solution:

```
x = -4:0;
y = x.^2 + 4.*x + 3;

plot(x,y, 'o-')
grid
```

Which gives the following plot:



Then we can easily see the solution from the plot, i.e., $y = 0$ when $x = -3$ and $x = -1$.

[End of Example]

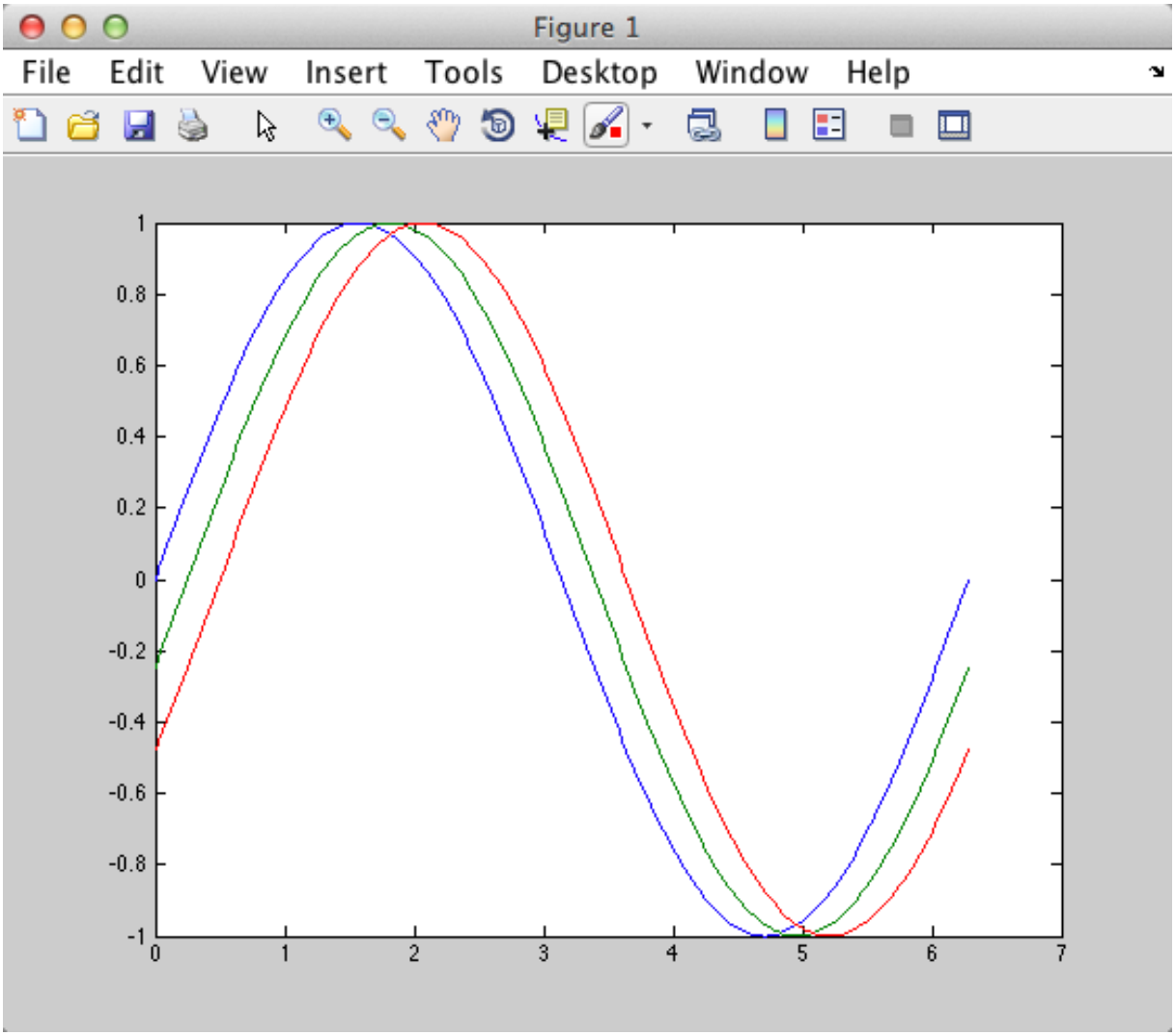
7.1 Plotting Multiple Data Sets in One Graph

In MATLAB it is easy to plot multiple data set in one graph.

Example:

```
x = 0:pi/100:2*pi;
y = sin(x);
y2 = sin(x-.25);
y3 = sin(x-.5);
plot(x,y, x,y2, x,y3)
```

This gives the following plot:



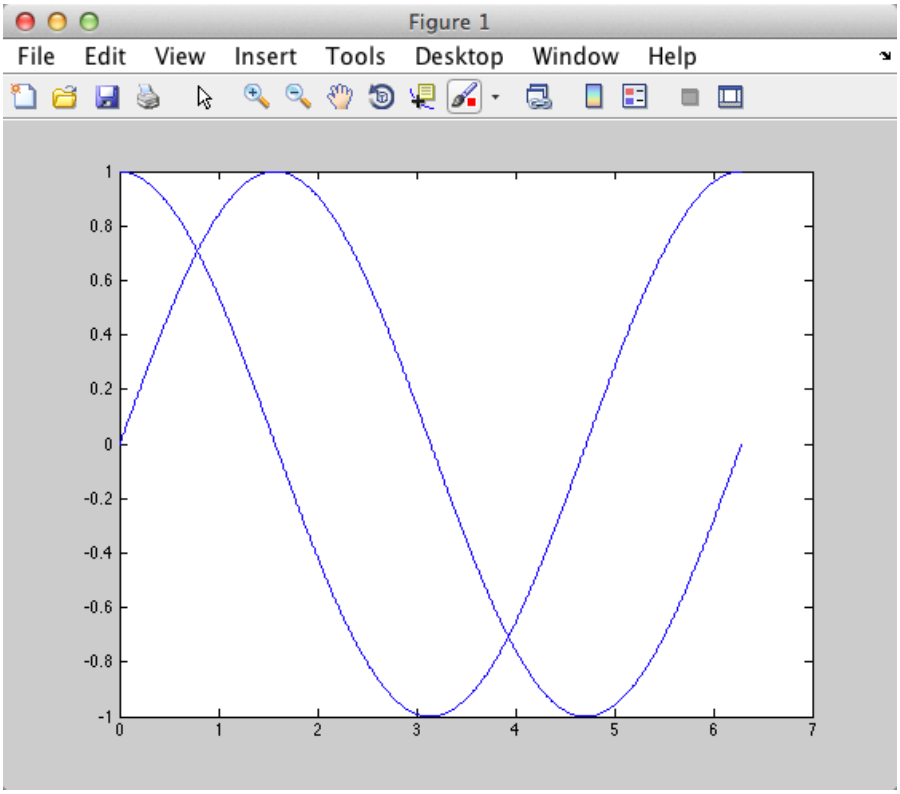
Another approach is to use the **hold** command:

```
x=0:0.01:2*pi;

plot(x, sin(x))
hold on

plot(x, cos(x))
hold off
```

This gives the following plot:



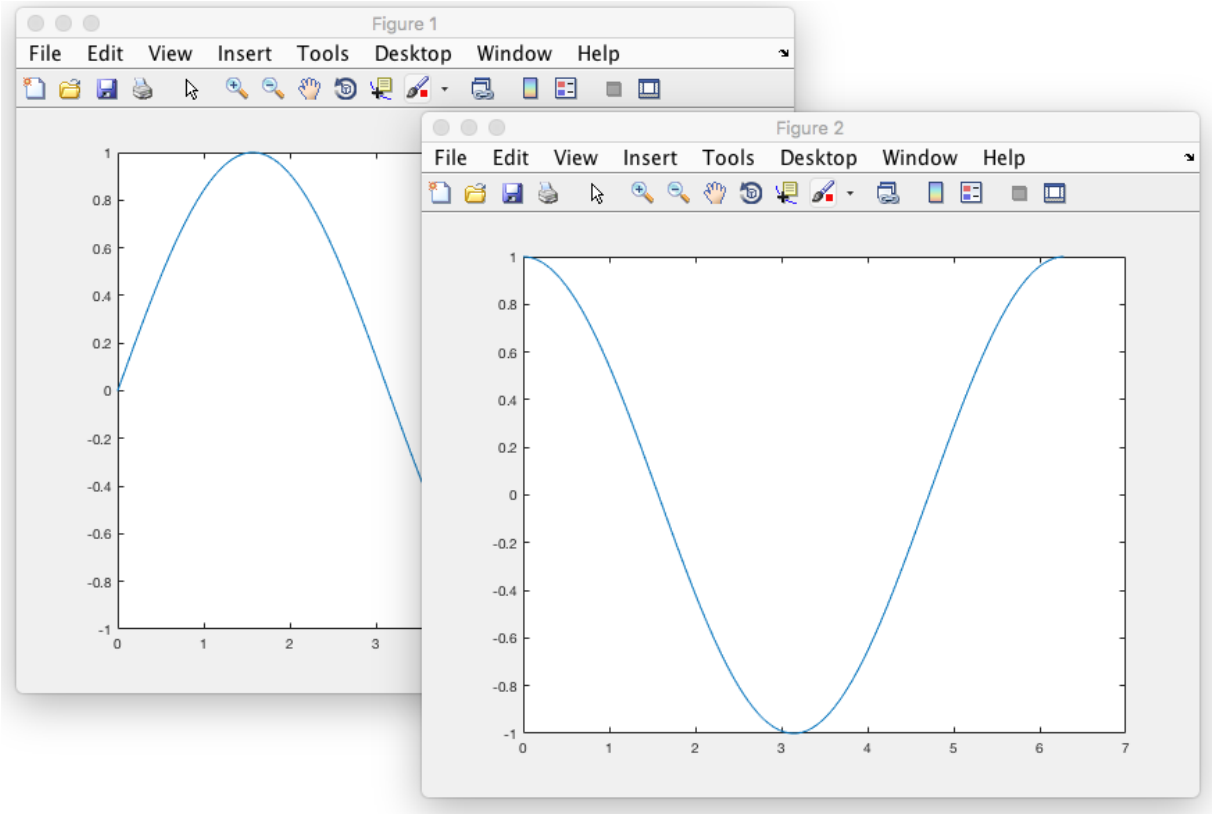
[End of Example]

You can also do the plotting in different plots using the **figure()** command.

Example:

```
x=0:0.01:2*pi;
figure(1)
plot(x, sin(x))
figure(2)
plot(x, cos(x))
```

The results will be like this:



[End of Example]

Task 10: Plot of dynamic system

Given the following differential equation:

$$\dot{x} = ax$$

where $a = -\frac{1}{T}$,where T is the time constant

The solution for the differential equation is:

$$x(t) = e^{at}x_0$$

Set $T = 5$ and the initial condition $x(0) = 1$

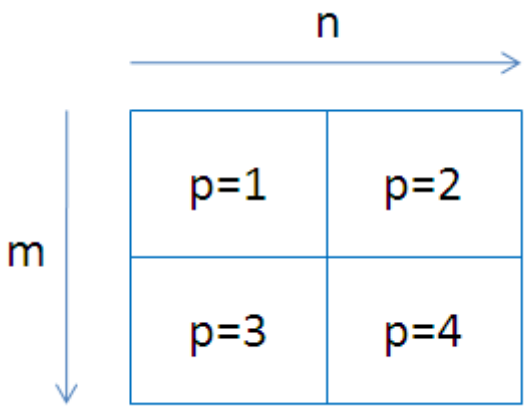
→ Create a Script in MATLAB (.m file) where you plot the solution $x(t)$ in the time interval $0 \leq t \leq 25$

→ Add Grid, and proper Title and Axis Labels to the plot.

[End of Task]

Displaying Multiple Plots in one Figure – Sub-Plots

The subplot command enables you to display multiple plots in the same window or print them on the same piece of paper. Typing "subplot(m,n,p)" partitions the figure window into an m-by-n matrix of small subplots and selects the pth subplot for the current plot. The plots are numbered along the first row of the figure window, then the second row, and so on.



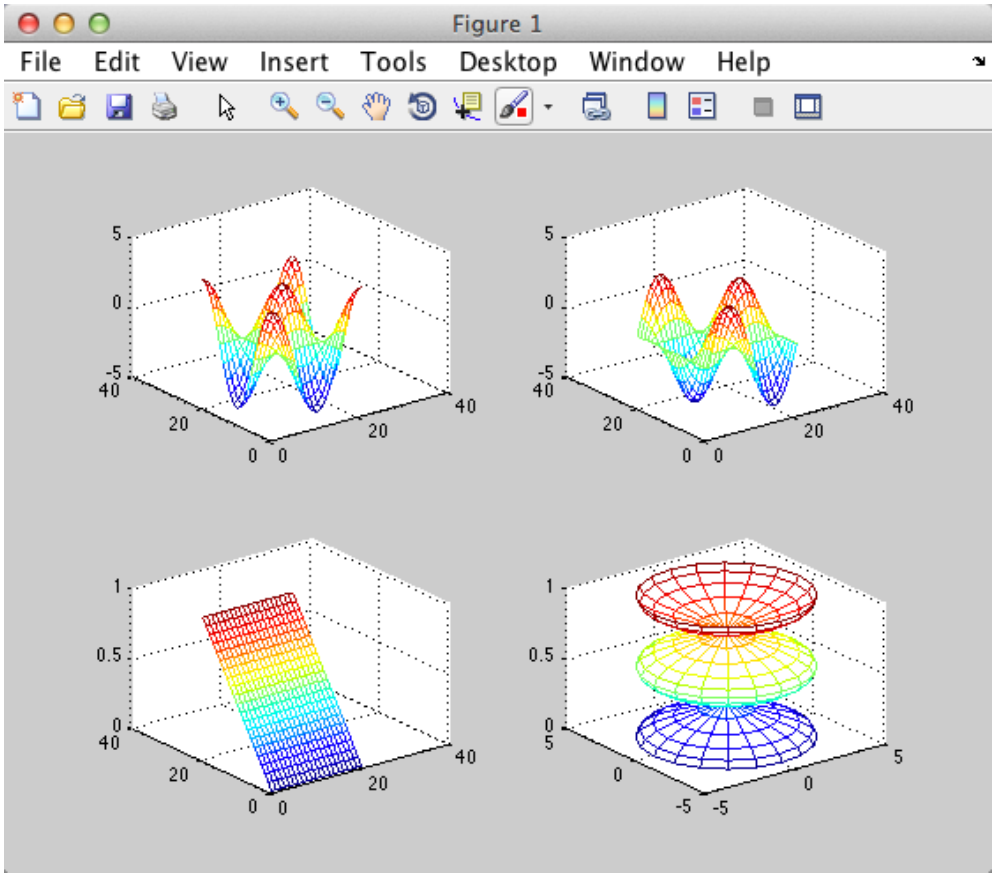
The syntax is as follows:

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

Example: Subplots

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```

This gives:



[End of Example]

Task 11: Sub-plots

Plot Sin(x) and Cos(x) in 2 different subplots.

Add Titles and Labels.

[End of Task]

Customizing

There is lots of customizing you can do with plots, e.g., you can add a title, x- and y-axis labels, add a legend and customize line colors and line-styles.

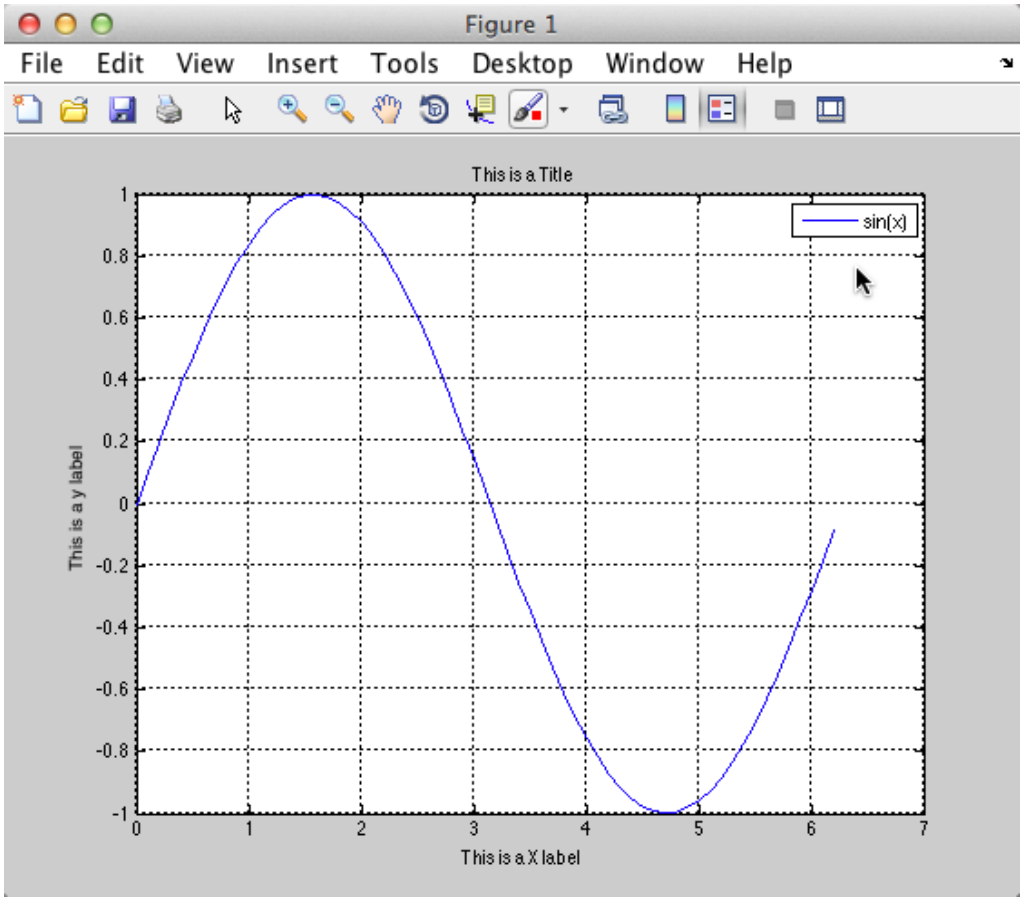
The functions for doing this is; **title**, **xlabel**, **ylabel**, **legend**, etc.

Example:

```
x=0:0.1:2*pi;
plot(x, sin(x))
```

```
%Customize the Plot:
title('This is a Title')
xlabel('This is a X label')
ylabel('This is a y label')
legend('sin(x)')
grid on
```

This gives the following plot:



[End of Example]

For line colors and line-styles we have the following properties we can use for the **plot** function:

Line Styles:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

Marker specifiers:

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or s	Square
'diamond' or d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'pentagram' or p	Five-pointed star (pentagram)
'hexagram' or h	Six-pointed star (hexagram)

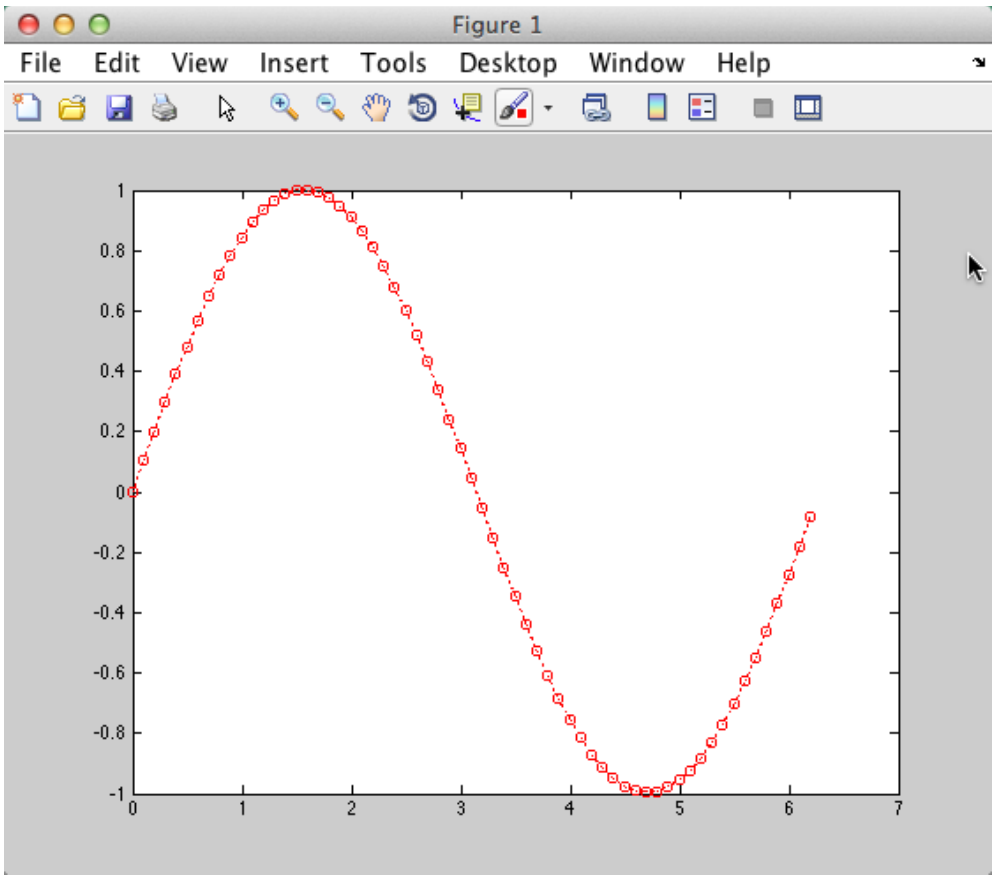
Colors:

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

Example:

```
>> x=0:0.1:2*pi;
>> plot(x, sin(x), 'r:o')
```

This gives the following plot:



[End of Example]

Other Plots

MATLAB offers lots of different plots.

Task 12: Other Plots

Check out the help for the following 2D functions in MATLAB: loglog, semilogx, semilogy, plotyy, polar, fplot, fill, area, bar, barh, hist, pie, errorbar, scatter.

→ Try some of them, e.g., bar, hist and pie.

[End of Task]