Department of Electrical Engineering, Pulchowk Campus*, Tribhuwan University, Nepal*

A Series of Module has been designed and developed by author mentioned above with full copyright for a course of **Control System** to demonstrante the concept of control system and its implementation in real life to second year Aerospace engineering students at Tribhuwan University.

MODULE I: Introduction to MATLAB

Table of Contents

1.1 Introduction

MATLAB is a very powerful tool developed by MATLAB Inc. used by Engineers of all domaind to design, visulaize and test the system developed by them.

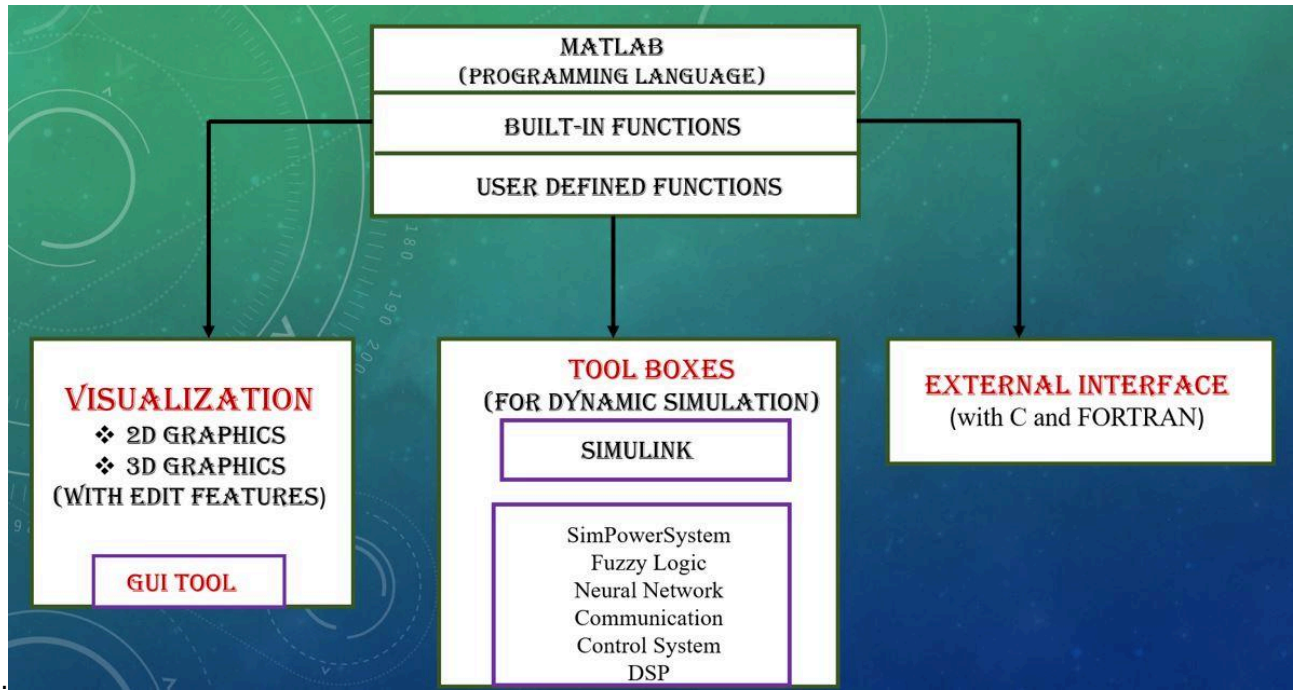
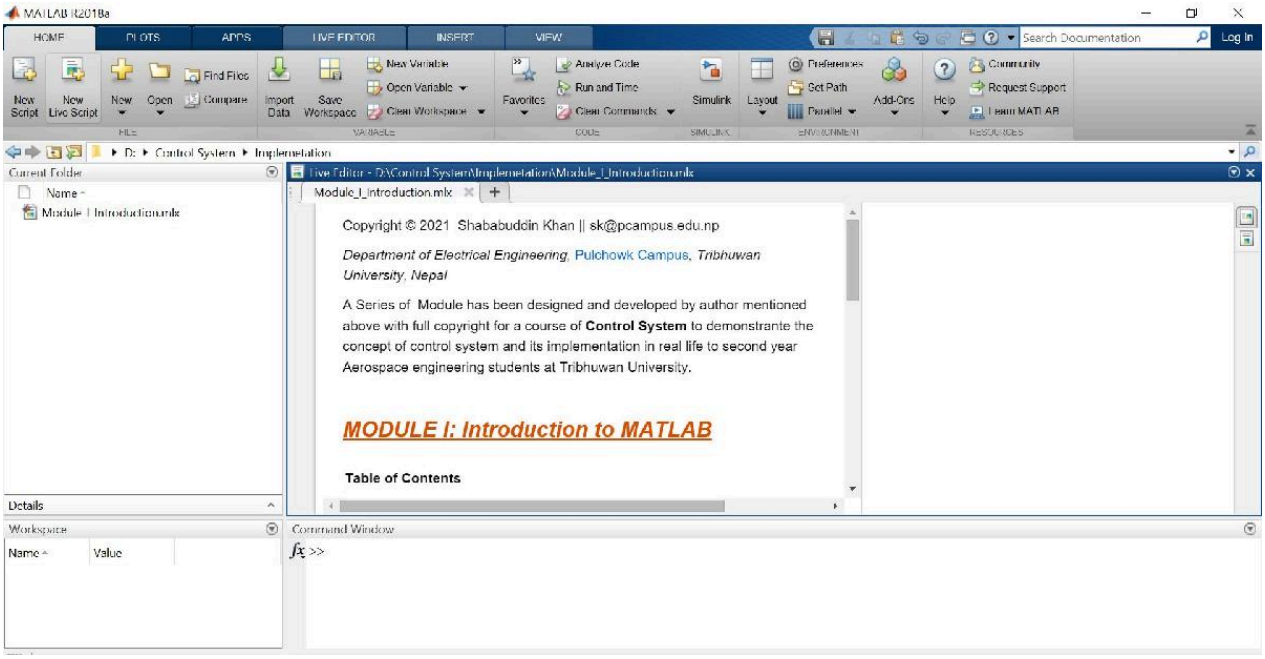


Fig: Block Representation of features of MATLAB[]

1.2 MATLAB Editor



1.3 Color Code in MATLAB

The MATLAB Edit window codes default colors:

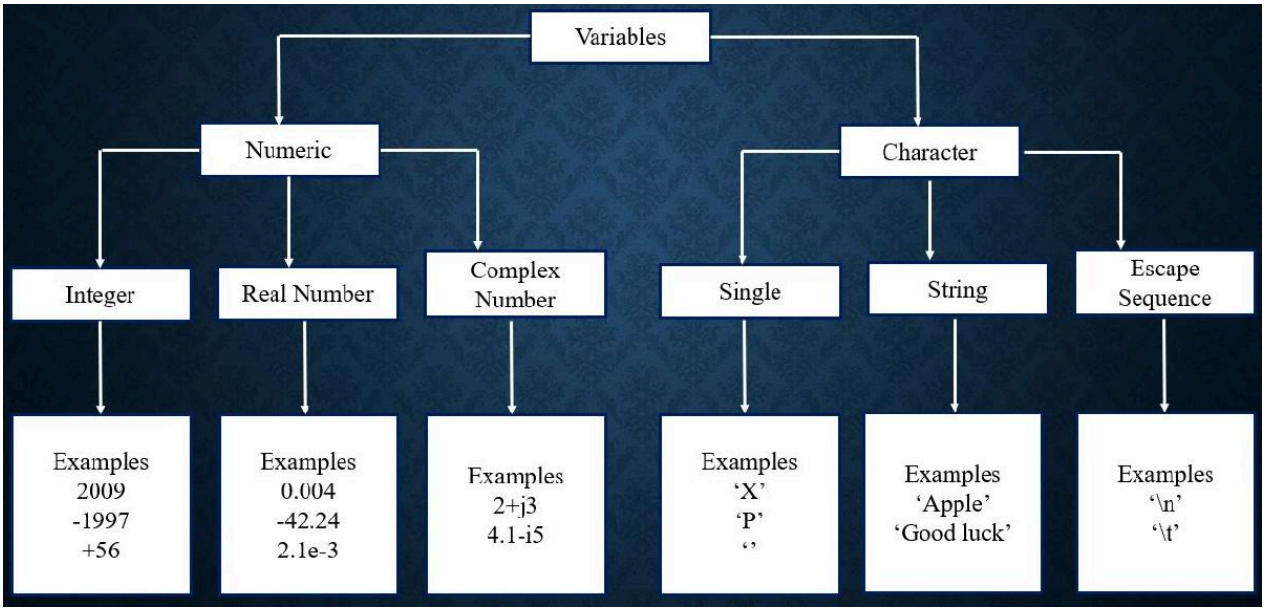
@ Comments(%) Green @Numbers and variables **Black**

@ Character string Violet @Language keywords(if, else, etc) Blue

```
% Comments are Green
n1=1;                                   % Number and Variables are Black
if(n1==1)                               % Language Keyword is blue
fprintf("Output =%d",n1);               % Language Keyword is violet
end
```

```
Output =1
```

1.4 Variable




```
% Variable Declaration is not necessary
%Variable can be used directly
% Naming of variable must follow same rule as in C Programming
```

1.5 Assigning Value to Variable

Storing Single Data

```
%Integer Variable
an=5;
%Float Variable
b=2.5;
%Chacter/String Variable
c='sk';
```

Storing Vector Data

```
%Row Vector
```

```
x1=[5 1 0.5]
```

```
x1 = 1×3
    5.0000    1.0000    0.5000
```

```
x2=[6,3,4]
```

```
x2 = 1×3
     6     3     4
```

```
% Column Vector
y1=[1;2;3.5]
```

```
y = 3×1
    1.0000
    2.0000
    3.5000
```

Storing Matrix Data

```
%Row Vector
m1=[5 1 0.5;1 2 4]
```

```
m1 = 2×3
    5.0000    1.0000    0.5000
    1.0000    2.0000    4.0000
```

```
% Column Vector
m2=[x1;x2]
```

```
m2 = 2×3
    5.0000    1.0000    0.5000
    6.0000    3.0000    4.0000
```

Creating Long date[use of :]

```
%For creating variable having data from 0 to 20 with interval of 1 [By Default]
d1=0:20
```

```
d1 = 1×21
     0     1     2     3     4     5     6     7     8     9    10    11    12 ···
```

```
dt=0.1;
%For creating variable having data from 0 to 20 with interval of 1
d2=0:dt:20
```

```
d2 = 1×201
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000 ···
```

1.6 Input Statements

By Slider

```
A=-10
```

```
A = -10
```

By Dropbox

```
B=3
```

```
B = 3
```

1.7 Output Statements

```
% Use of disp() Function
disp(A);
```

```
    70
```

```
% Use of fprintf() Function
fprintf('The value of B is %d',B);
```

```
The value of B is 3
```

1.8 Inbuilt Functions

- sin(x) Calculates sin(x) where x is in radian
- sind(x) Calculates sin(x) where x is in degree
- asin(x) Calculates inverse of sin(x) and gives the result in radian.
- asind(x) Calculates inverse of sin(x) and gives the result in degree.
- sqrt(x) Calculate square-root of x
- round(x) Round towards nearest integer
- **rem(x,y)** Gives remainder after division
- size(x) Gives Size of thematrix x
- length(x) Gives the length of the matrix or vector
- real(x) Gives real part from a complex number x
- img(x) Gives imaginary part from a complex number x
- conj(x) Find conjugate of complex number x
- abs(x) Absolute value of x/Magnitude of complex number x
- angle(x) Returns the phase angle in radians

```
% Inbuilt Function Example
angle=30;
aa=sin(angle)
```

```
aa = -0.9880
```

```
ab=sind(angle)
```

```
ab = 0.5000
```

1.9 Conversion in Matlab

```
p=5,q=10
```

```
p = 5
q = 10
```

```
%Converts Cartesian to polar form
%cart2pol
[theta, m] = cart2pol(p,q)
```

```
theta = 1.1071
m = 11.1803
```

```
%Converts polar to Cartesian form
%pol2cart
[p, q] = pol2cart(theta, m)
```

```
p = 5.0000
q = 10
```

```
% We also have conversion in data types
%num2str
%str2num
```

1.10 Operators

Arithmetic Operator

SYMBOL	OPERATION	EXAMPLE
+	Addition	9+7=16
-	Subtraction	9.1-7=2.1
*	MULTIPLICATION	2*4=8
/	Division(Right)	8/3=2.6667
\	Division(Left)	8/3=0.375 [=3/8]
^	Exponent	2^3 = 8 [=2^3]

Assignment Operator

Operation	Symbol
Equal	==
Not Equal	~=
Less than	<
Less than or equal	<=
Greater than	>
Greater than or equal	>=

Logical Operator

Operation	Symbol
and	&
or	|
not	~

1.11 Control Logic

if Statement

```
a=2;
if a==1   % Paranethesis is not necessary and { too.
    fprintf('Output is %d',a);
end
```

if - else Statement

```
a=6;
if mod(a,2)==0
        fprintf('Even Number');
else
        fprintf('Odd Number');
end
```

```
Even Number
```

if - elseif - else Statement

```
a=-2;
if a>0
        fprintf('Number is Positive');
elseif a<0
        fprintf('Number is Negative');
else
    fprintf('Number is Zero');
end
```

```
Number is Negative
```

Switch Statement

```
a=3;
switch(a)
    case 1              % No column required
        fprintf('One'); % No Break Required
    case 2
        fprintf('Two');
    case 3
        fprintf('Three');
    otherwise    % Default in C is replaced by Otherwise
        fprintf('Invalid Input');
end
```

```
Three
```

for loop

```
for i=1:1:10             %Start:Increment:Stop
    fprintf('%d\t',i)
end
```

```
1 2 3 4 5 6 7 8 9 10
```

while loop

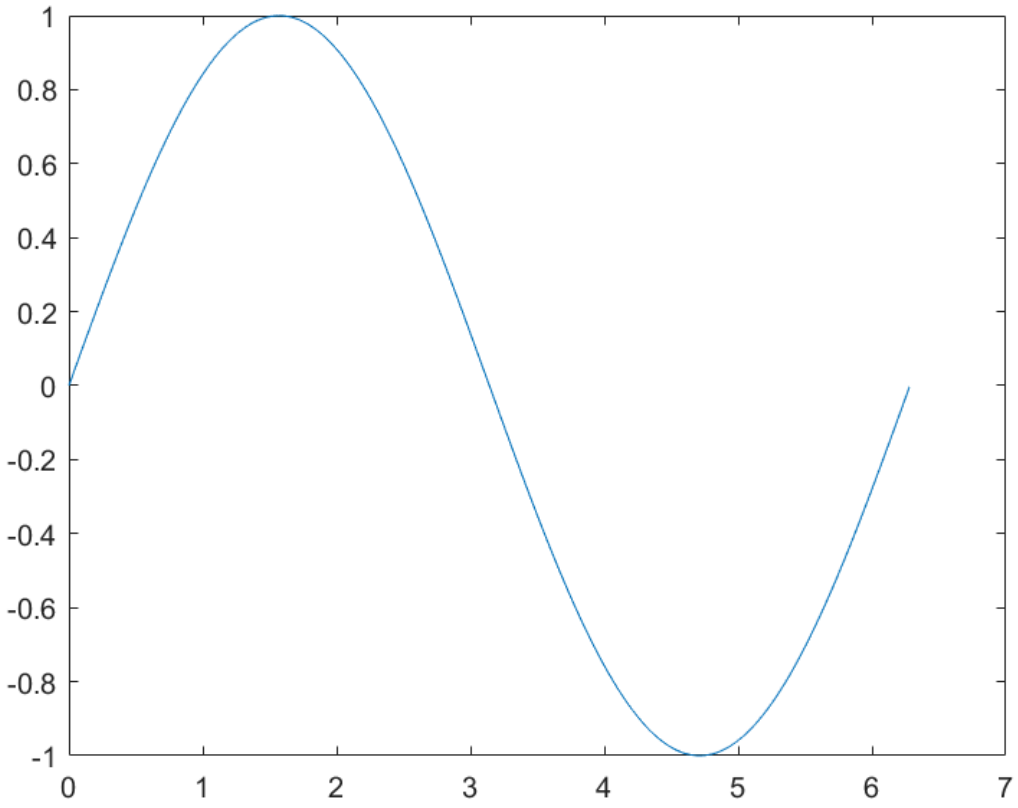
```
n=5;
while n>0
    fprintf('%d\t',n);
    n=n-1;
end
```

```
5 4 3 2 1
```

1.12 Data Visualization

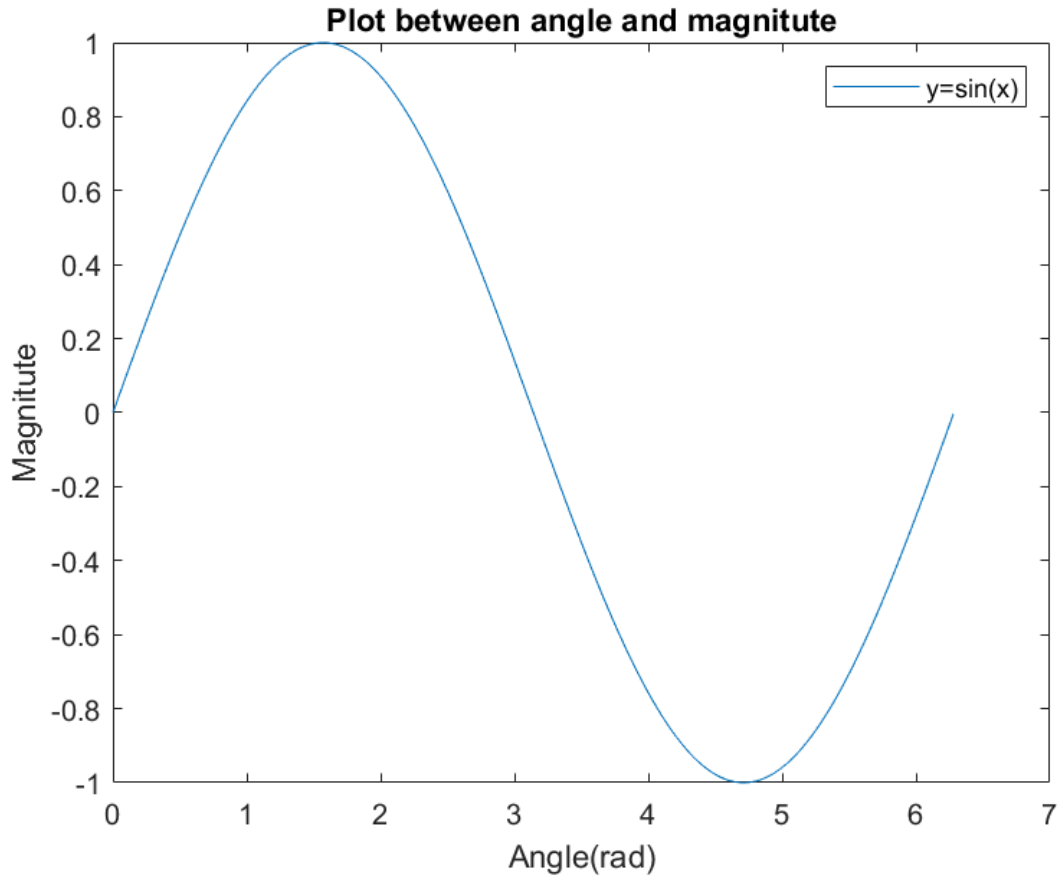
Single 2D Plot

```
x=0:0.01:2*pi;
y=sin(x);
%ploting 2D Graph
plot(x,y);
```



Single 2D Plot with its necessary information

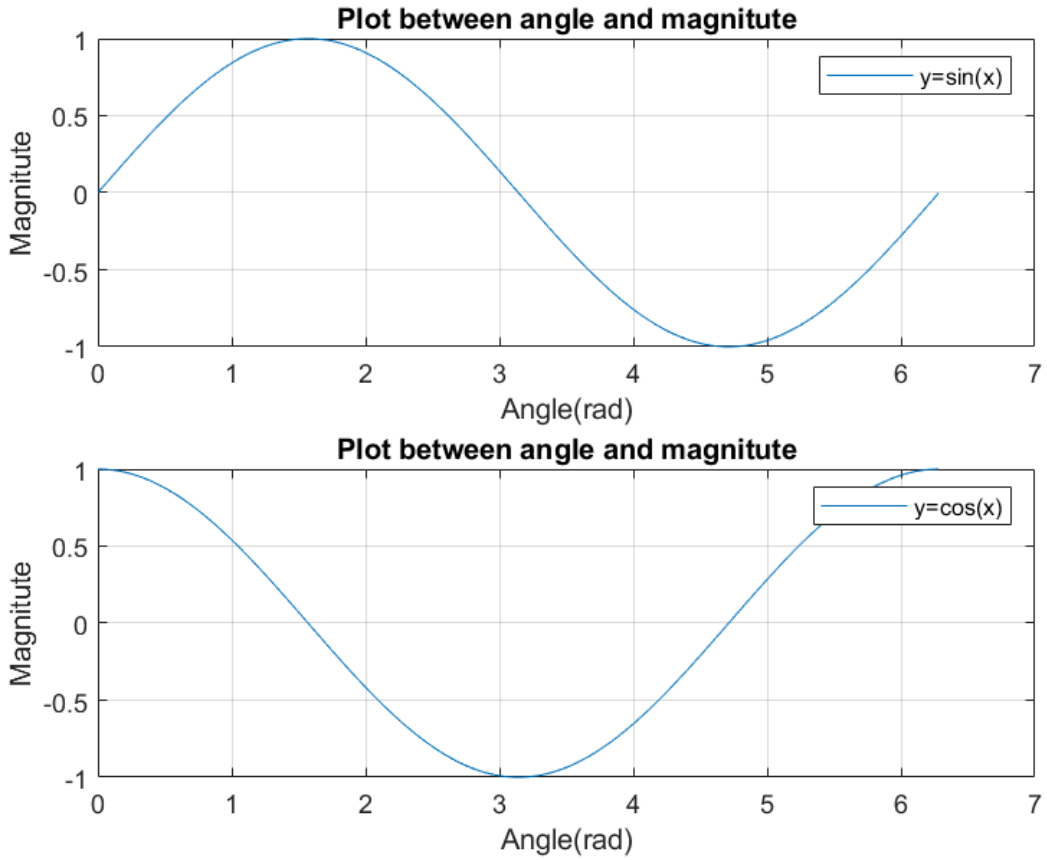
```
x=0:0.01:2*pi;
y=sin(x);
%ploting 2D Graph
plot(x,y);
xlabel('Angle(rad)');
ylabel('Magnitute');
title('Plot between angle and magnitute');
%grid on;
legend('y=sin(x)');
```



Two 2D Plot using subplot()

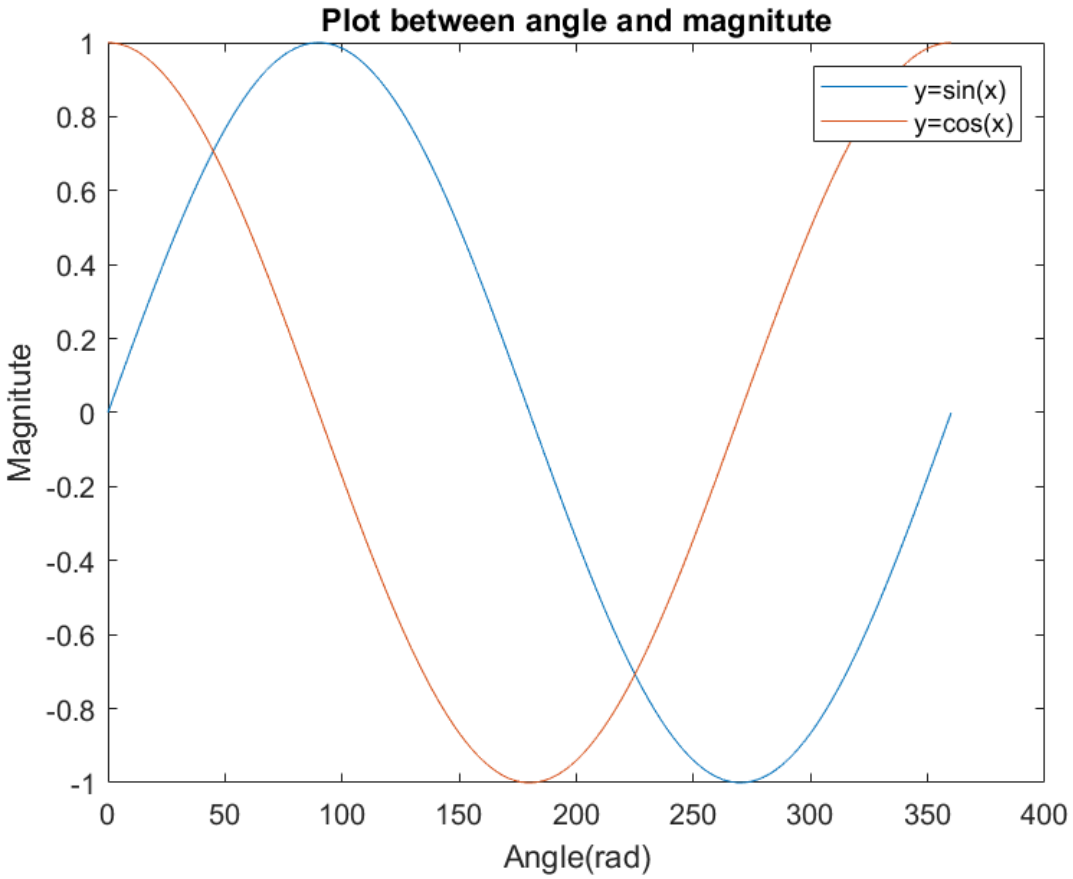
```
%ploting 2D Graph of sin(x)
subplot(2,1,1);             %No.of row, No. of Column, Position
x=0:0.01:2*pi;
y=sin(x);
plot(x,y);
xlabel('Angle(rad)');
ylabel('Magnitute');
title('Plot between angle and magnitute');
grid on;
legend('y=sin(x)');

%ploting 2D Graph of cos(x)
subplot(2,1,2);             %No.of row, No. of Column, Position
x=0:0.01:2*pi;
y=cos(x);
plot(x,y);
xlabel('Angle(rad)');
ylabel('Magnitute');
title('Plot between angle and magnitute');
grid on;
legend('y=cos(x)');
```



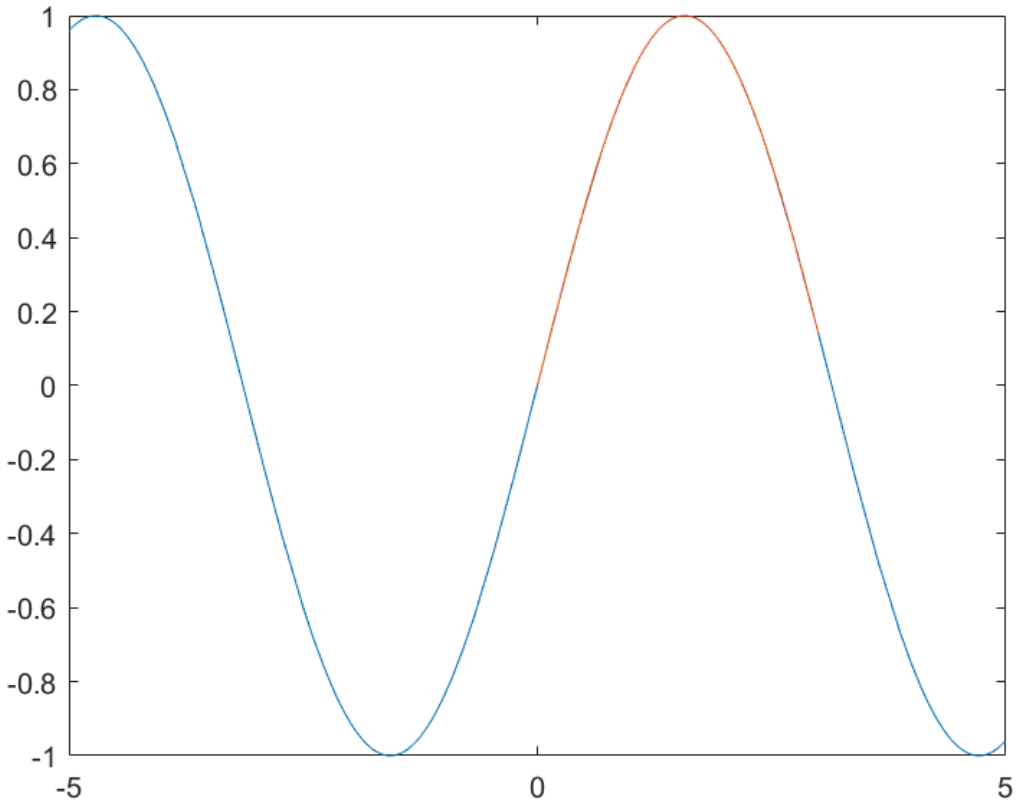
Two 2D Plot in same graph

```
%hold on/off is used to
x=0:0.01:360;
y=sind(x);
z=cosd(x);
% figure; %For taking new figure
plot(x,y);
%Hold Previous Plot
hold all;
plot(x,z);
hold off;
legend('y=sin(x)','y=cos(x)');
xlabel('Angle(rad)');
ylabel('Magnitute');
title('Plot between angle and magnitute');
```



2D Plot using fplot()

```
%fplot is used to plot a function within a time interval [-5,5] by default
% or we can mention the time limit
%x=0:0.01:360;
y=@(x) sin(x);     %defining a function
fplot(y);
hold on;
fplot(y,[0 3]);  % for range 0 to 3
hold off;
```



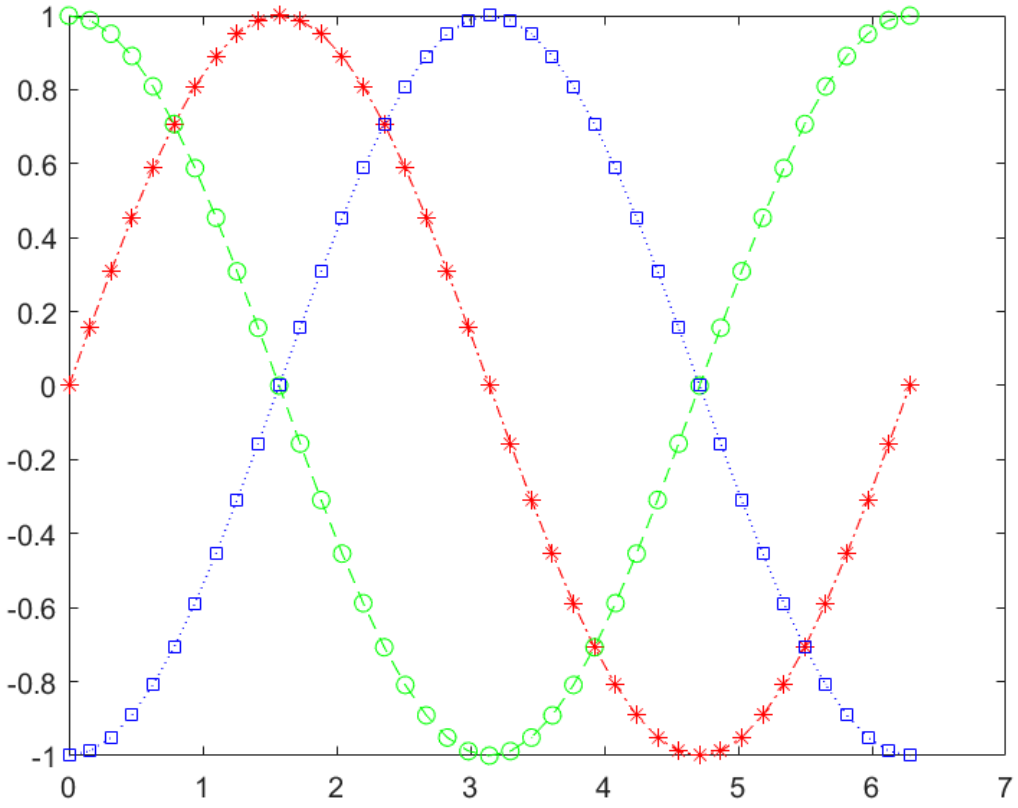
Adding Features to Plot

Syntax:

plot(x,y, 'line specifier', 'property name', 'property value');

Line Style		Marker Style		Color	
-	Solid	-	Point	**y**	yellow
:	Dotted	**o**	Circle	**m**	magenta
-.	Dash-dot	**x**	X-mark	**c**	cyan
--	Dashed	+	Plus	**r**	red
<none>	No Line	*	Star	**g**	green
		s	Square	**b**	blue
		d	Diamond	**w**	white
		v	Triangle(down)	**k**	black
		^	Triangle(up)		
		<	Triangle(left)		
		>	Triangle(right)		
		p	Pentagram		
		h	Hexagram		
		<none>	No marker		

```
figure
t = 0:pi/20:2*pi;
plot(t,sin(t),'-.r*')
hold on
plot(t,cos(t),'--go')
%plot(t,cos(t+pi),':bs')
hold off
```



1.13 Laplace Transform

```
syms t;  % Symbolic Object i.e can be assigned
laplace (t) %Laplace
```

```
ans =
```

$$\frac{1}{s^2}$$

```
laplace(t/t) % Laplace of constant
```

```
ans =
```

$$\frac{1}{s}$$

1.14 Inverse Laplace Transform

```
syms s a;  % Symbolic Object i.e can be assigned
ilaplace (1/s^3) % Inverse Laplace Transform
```

```
ans =
```

$$\frac{t^2}{2}$$

```
ilaplace(a/(s+a))
```

ans = $a\,\mathrm{e}^{-a\,t}$

References

1. Mathworks.inc
2. S. Khan, N.K.Yadav, "A training Course of MATLAB", IOE, TU