Graphics – 2D and 3D plots

Plot continuous, discrete, surface, and volume data

Create a 2-D Line Plot

- Create a two-dimensional line plot using the plot function.
- Eg: plot the value of the sine function from 0 to 2π.

Plot

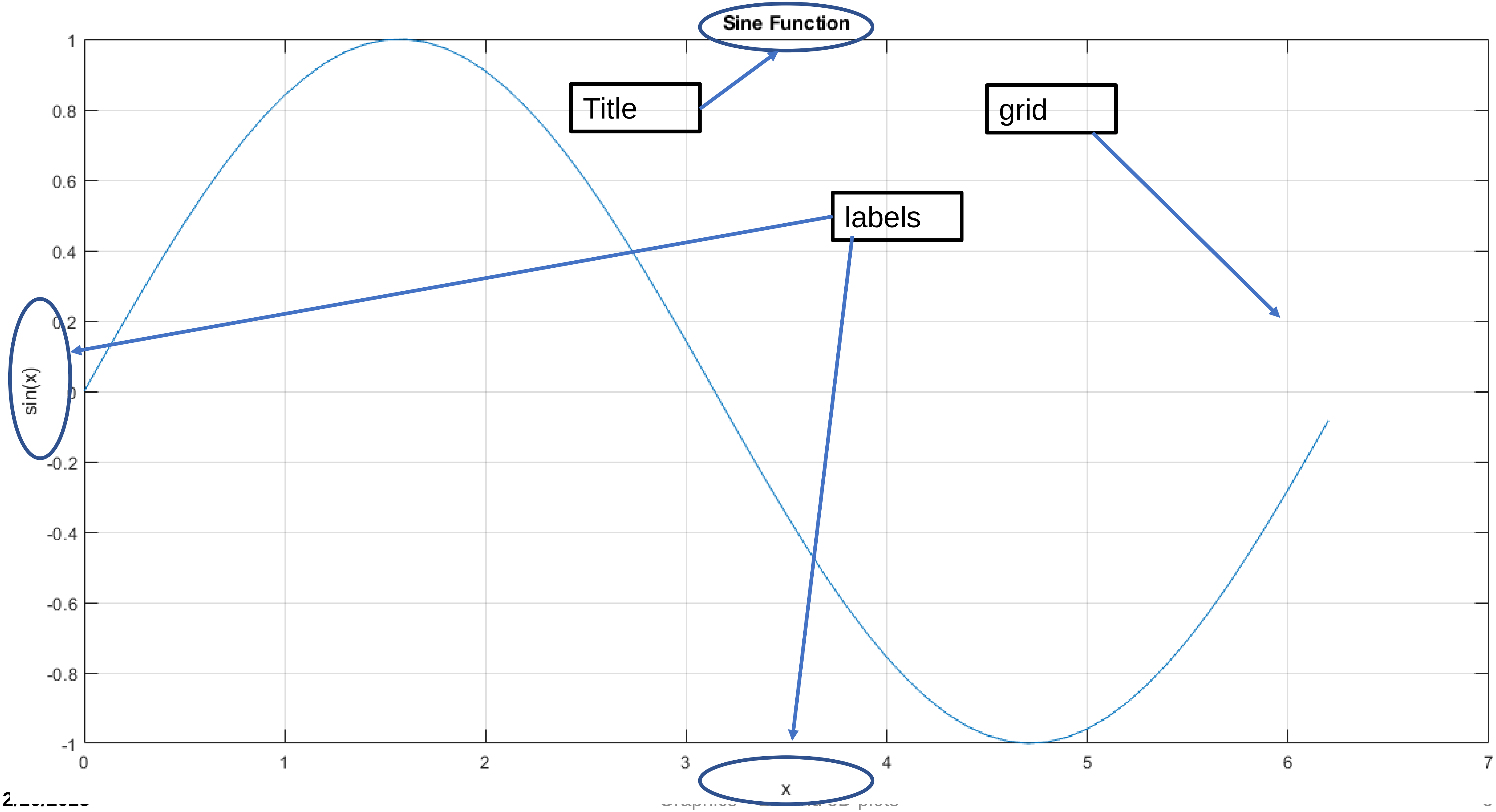
```
>> x = 0:0.1:2*pi;
 >> y = sin(x);
>> plot(x,y)
```

Label the axes and add a title

```
>>xlabel('x’)                    %add label
>>ylabel('sin(x)’)
>>title('Sine Function’)     %add title
```

Display the grid lines for a sine plot

```
>>grid on
```



Title

grid

labels

Multiple plots - using hold command

- By default, MATLAB **clears** the **figure before** each **plotting command**.
- Use the **figure** command to open a new figure window.
- Plot **multiple lines** using the **hold** on command.
- Until you use hold off or close the window, all plots appear in the current figure window.

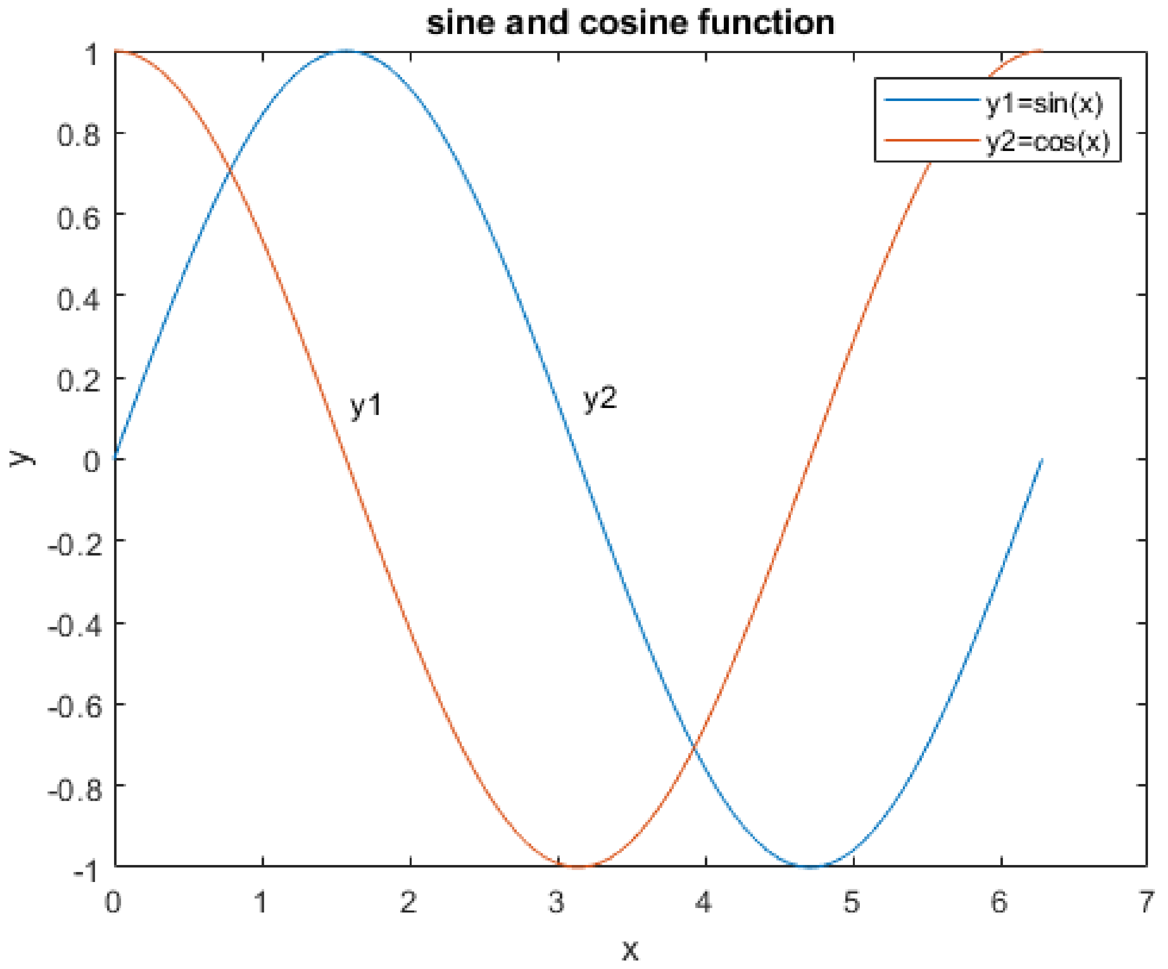
```
figure
x = linspace(0,2*pi,100);
y1 = sin(x);
plot(x,y)
hold on

y2 = cos(x);
plot(x,y2)
hold off


legend(‘y1= sin(x)’,’y2= cos(x)’);
```

Add Legend

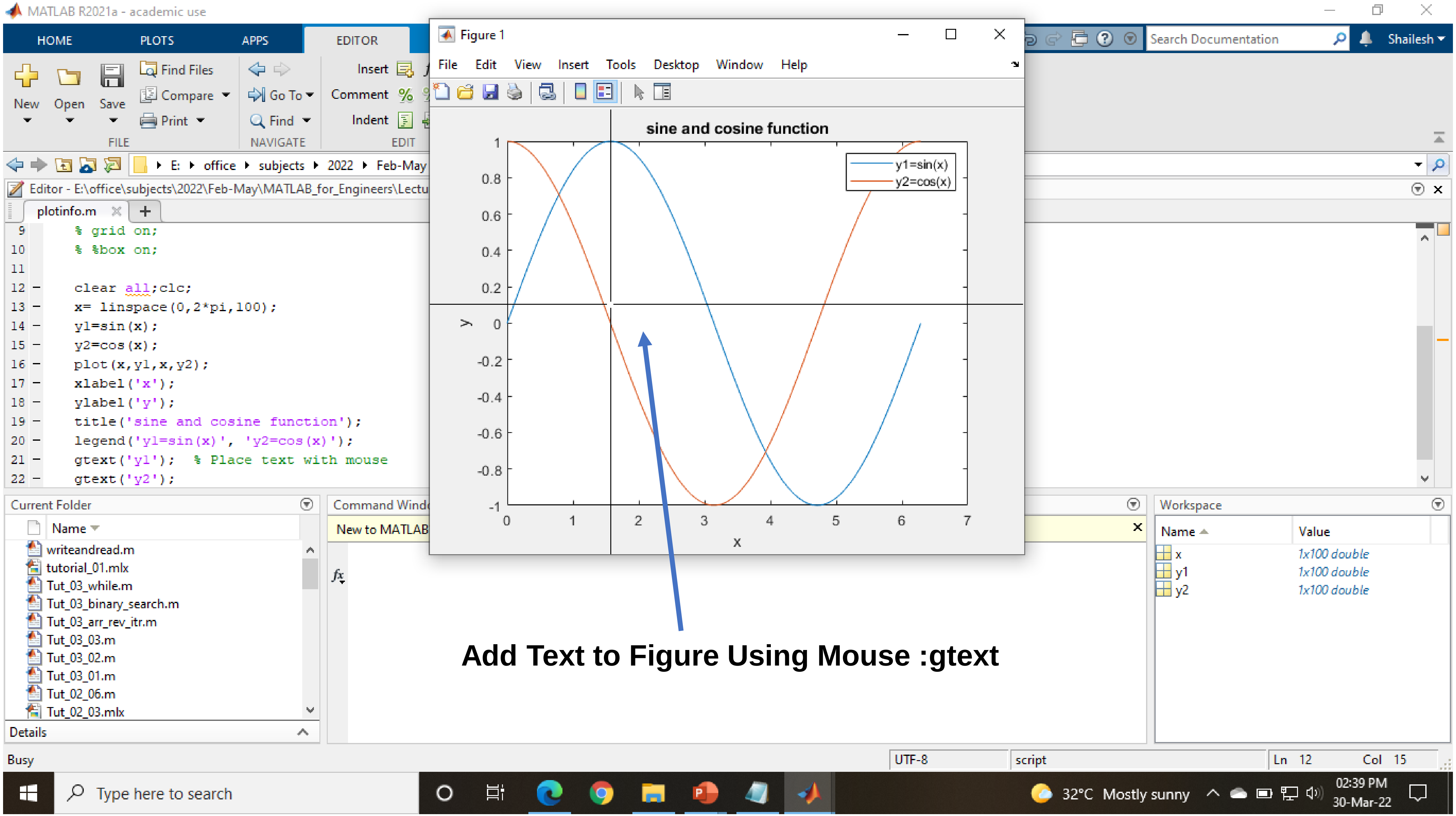
Add a legend to the graph that identifies each data set using the legend function

Use of plot command for Plotting Multiple Lines

```
>> x= linspace(0,2*pi,100)
>> y1=sin(x)
>> y2=cos(x)
>> plot(x,y1,x,y2)
>> xlabel('x')
>> ylabel('y')
>> title('sine and cosine function')
>> legend('y1=sin(x)', 'y2=cos(x)’)
>> gtext(‘y1’)             % Place text
with mouse
>> gtext(‘y2')
```



Add Text to Figure Using Mouse :gtext



```
    % grid on;
    % %box on;

    clear all;clc;
    x= linspace(0,2*pi,100);
    y1=sin(x);
    y2=cos(x);
    plot(x,y1,x,y2);
    xlabel('x');
    ylabel('y');
    title('sine and cosine function');
    legend('y1=sin(x)', 'y2=cos(x)');
    gtext('y1');  % Place text with mouse
    gtext('y2');
```

writeandread.m
tutorial_01.mlx
Tut_03_while.m
Tut_03_binary_search.m
Tut_03_arr_rev_itr.m
Tut_03_03.m
Tut_03_02.m
Tut_03_01.m
Tut_02_06.m
Tut_02_03.mlx

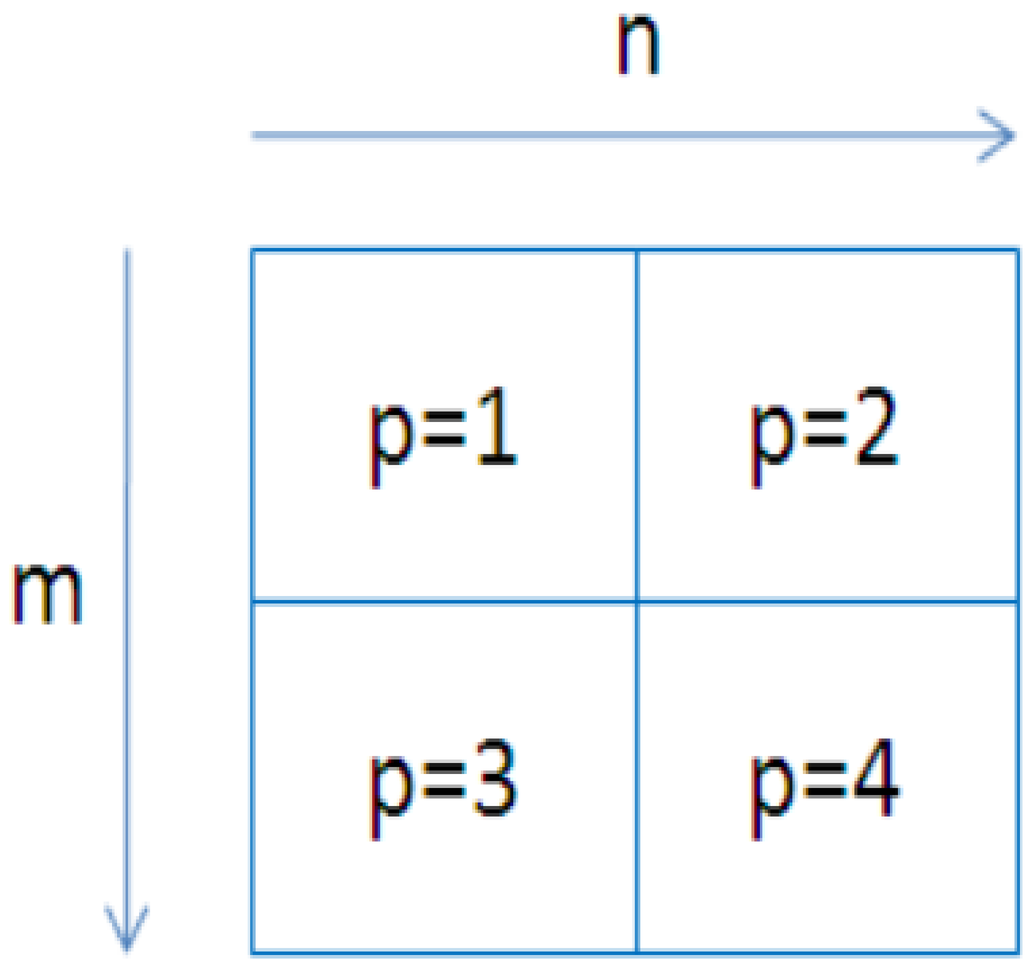
Name	Value
x	1x100 double
y1	1x100 double
y2	1x100 double

Add Text to Figure Using Mouse :gtext

Sub-plots

- Displaying **Multiple** Plots in **one** Figure – **Sub-Plots**

- **subplot**(m,n,p)

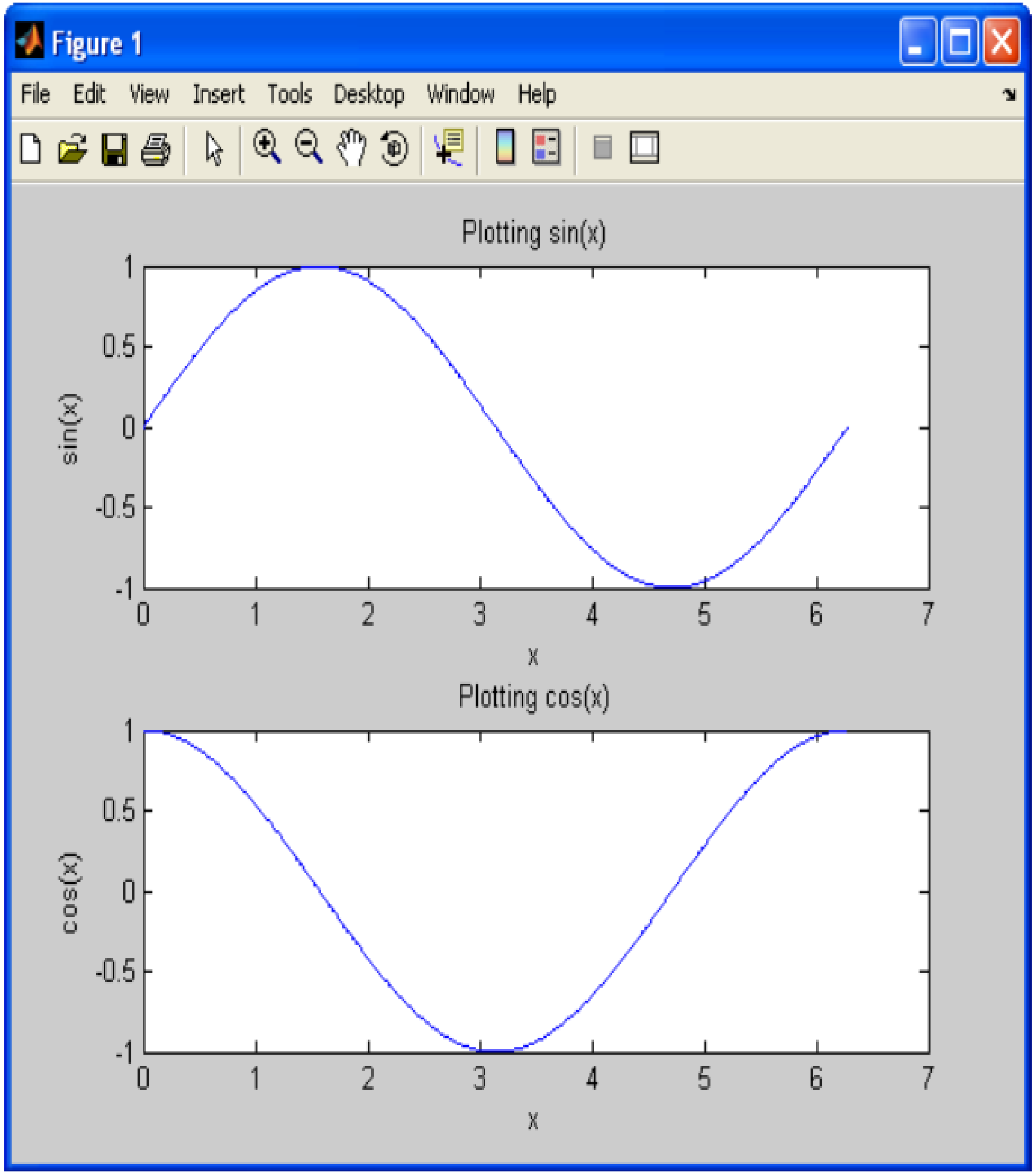


n

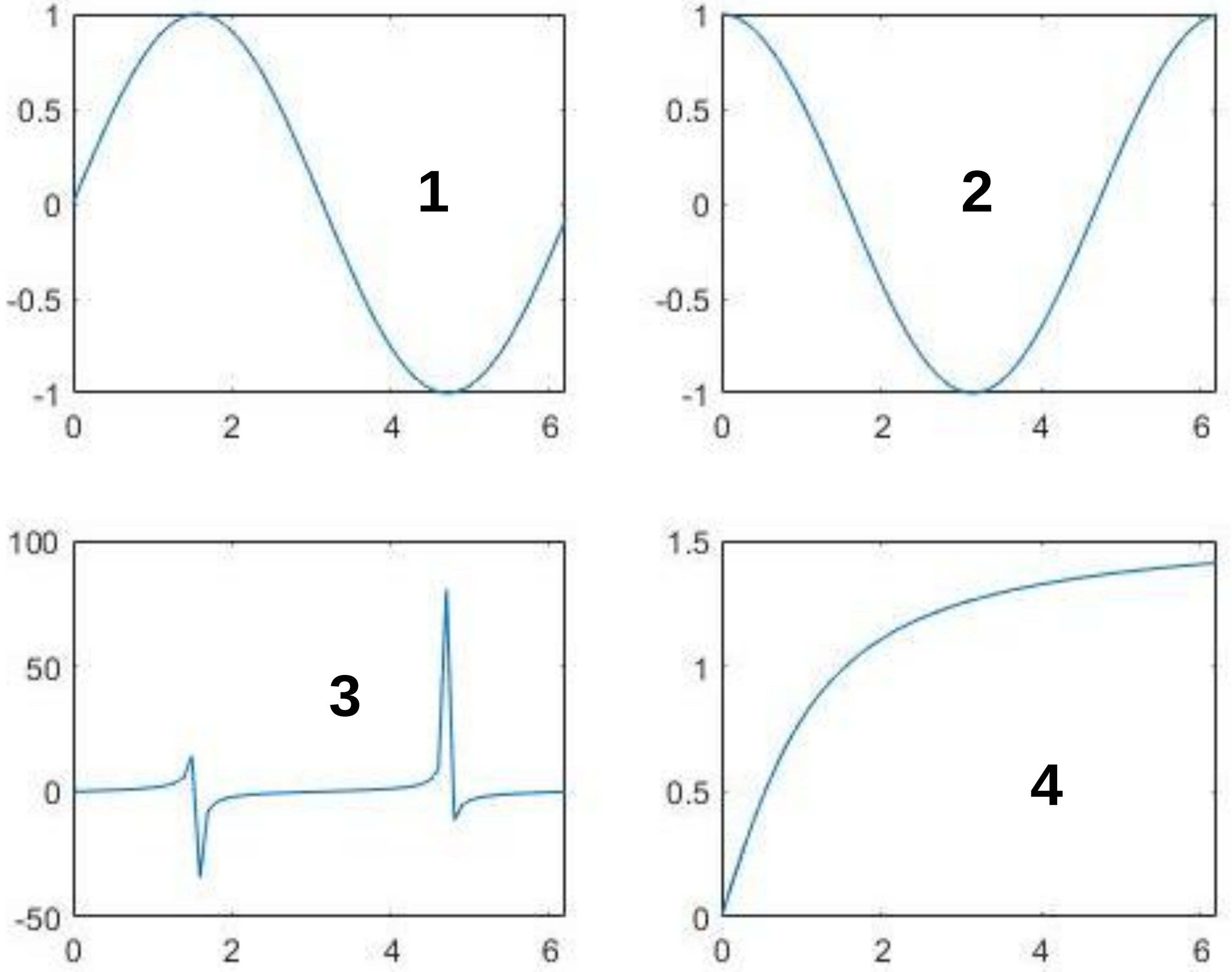
m

p=1	p=2
p=3	p=4

```
% Define x-values
x=0:0.01:2*pi;

% subplot 1
subplot(2,1,1) ;
plot(x, sin(x)) ;
title('Plotting sin(x)’) ;
xlabel('x’) ;
ylabel('sin(x)’) ;

% Subplot 2
subplot(2,1,2);
plot(x, cos(x));
title('Plotting cos(x)’) ;
xlabel(‘x’);
ylabel('cos(x)’) ;
```



```
x=0:0.1:2*pi;

y=sin(x);        y2=cos(x);
y3=tan(x);     y4=atan(x);

subplot(2,2,1) ; plot(x,y);
subplot(2,2,2); plot(x,y2);
subplot(2,2,3); plot(x,y3);
subplot(2,2,4); plot(x,y4);
```



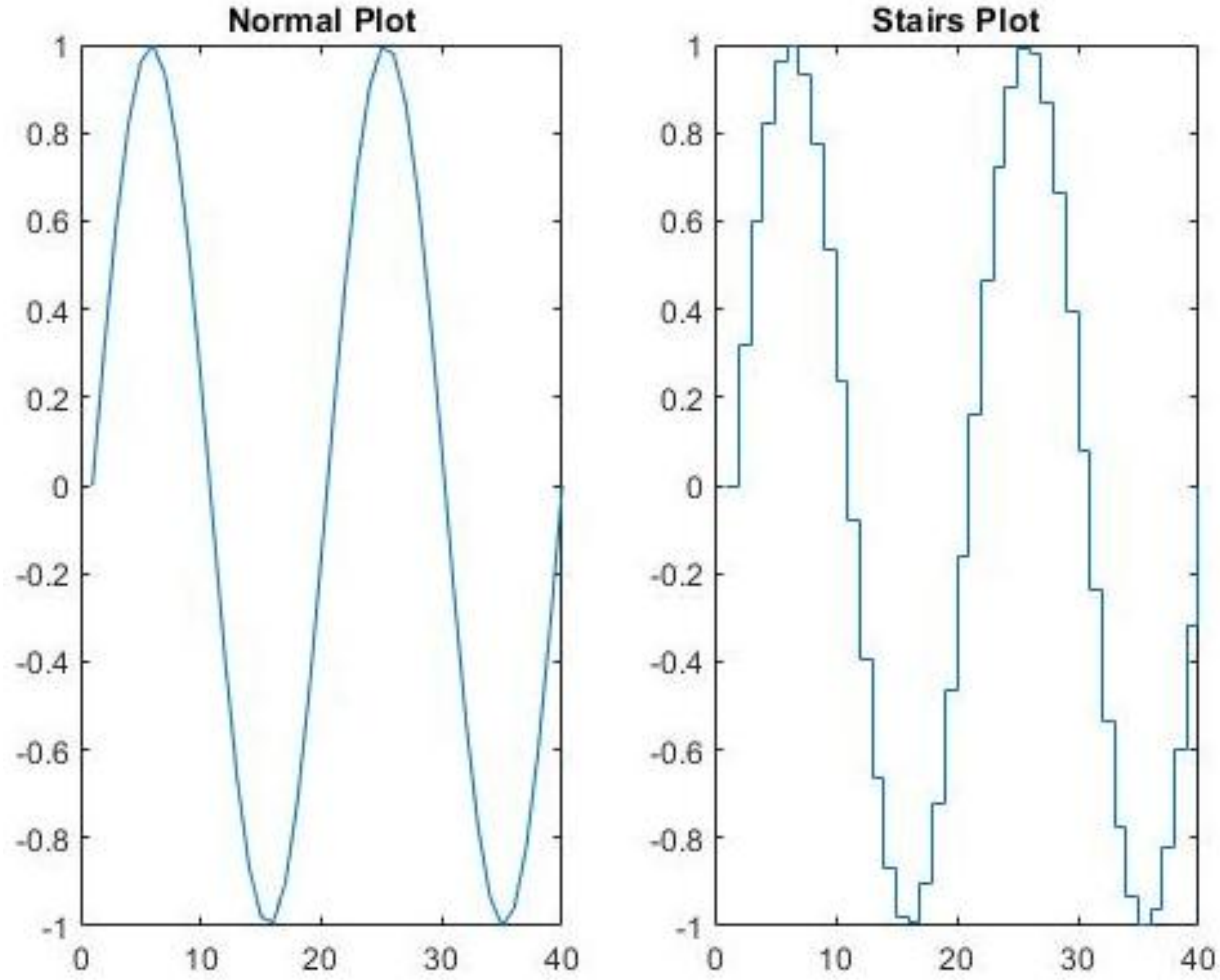
Special 2-D plots

plot	2-D line plot
plot3	3-D point or line plot
stairs	Stairstep graph
errorbar	Line plot with error bars
area	Filled area 2-D plot
stackedplot	Stacked plot of several variables with common x-axis

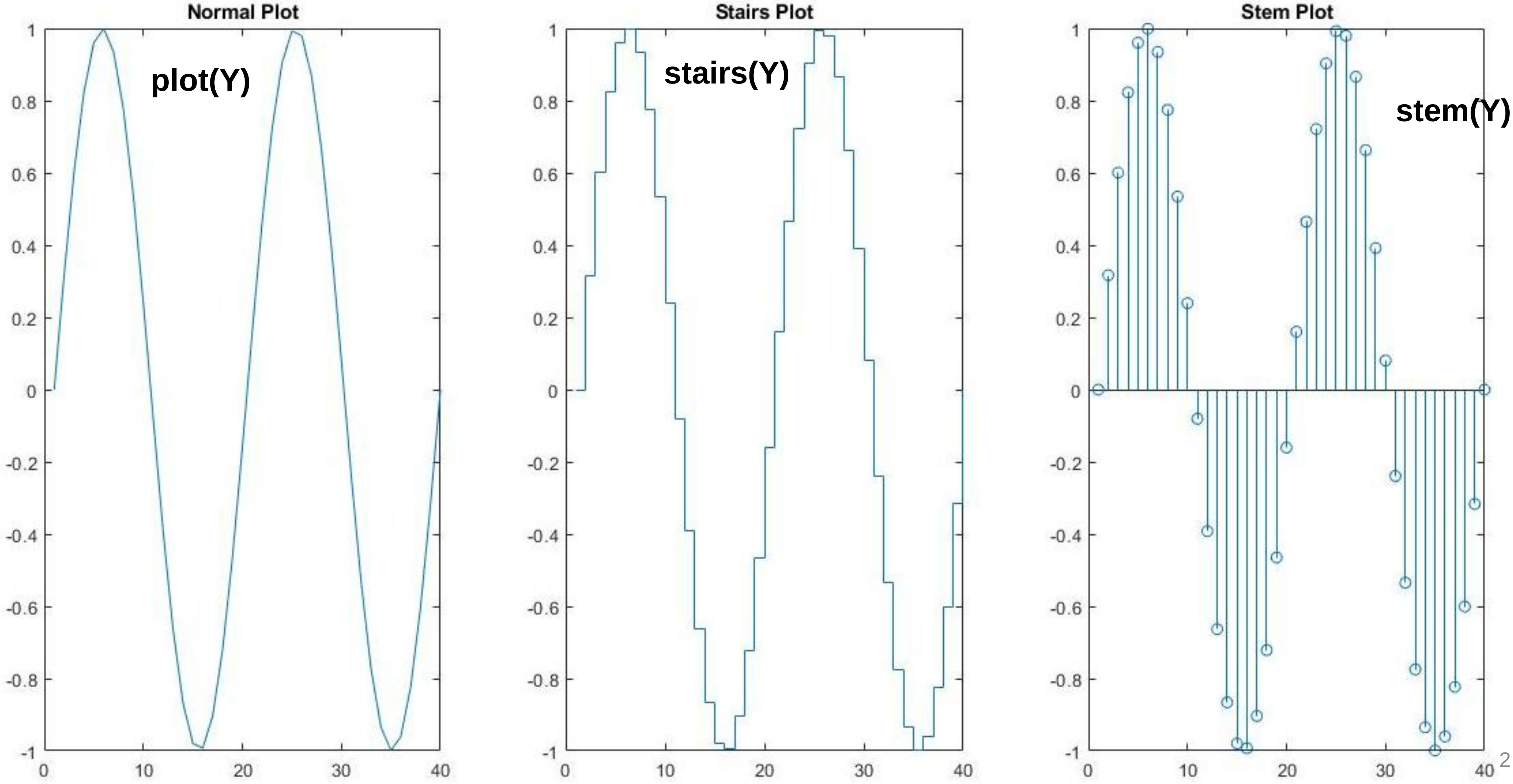
stairs

- Stairstep graph

```
X = linspace(0,4*pi,40);
Y = sin(X);

figure

subplot(1,2,1);
plot(Y);title('Normal Plot’);

subplot(1,2,2);
stairs(Y);title('Stairs Plot');
```



Plot a stem graph



plot(Y)

stairs(Y)

stem(Y)

Log plots

loglog	Log-log scale plot
semilogx	Semilog plot (x-axis has log scale)
semilogy	Semilog plot (y-axis has log scale)

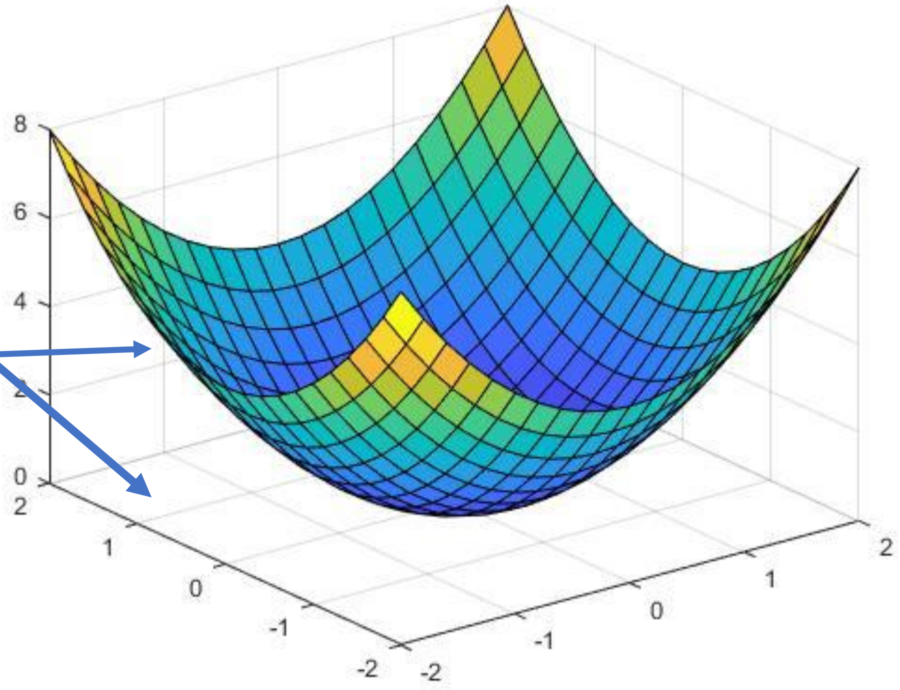
3-D plots

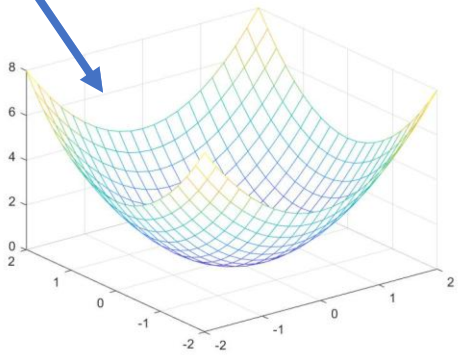
plot3	To plot simple 3D plot
bar3	To plot 3-D vertical bar graph
bar3h	To plot 3-D horizontal bar graph
pie3	To plot a 3-D pie plot
stem3	3-D stem plot
meshgrid	To give array for mesh grid
mesh	To plot wireframe mesh plot
surf	Surface plot
contour	To give contour of the given matrix
contour3	To give contour over the plane

Three-dimensional plots typically display a surface defined by a function in two variables, ***z = f(x,y)***

To evaluate *z*, first create a set of (*x*,*y*) points over the domain of the function using **meshgrid**

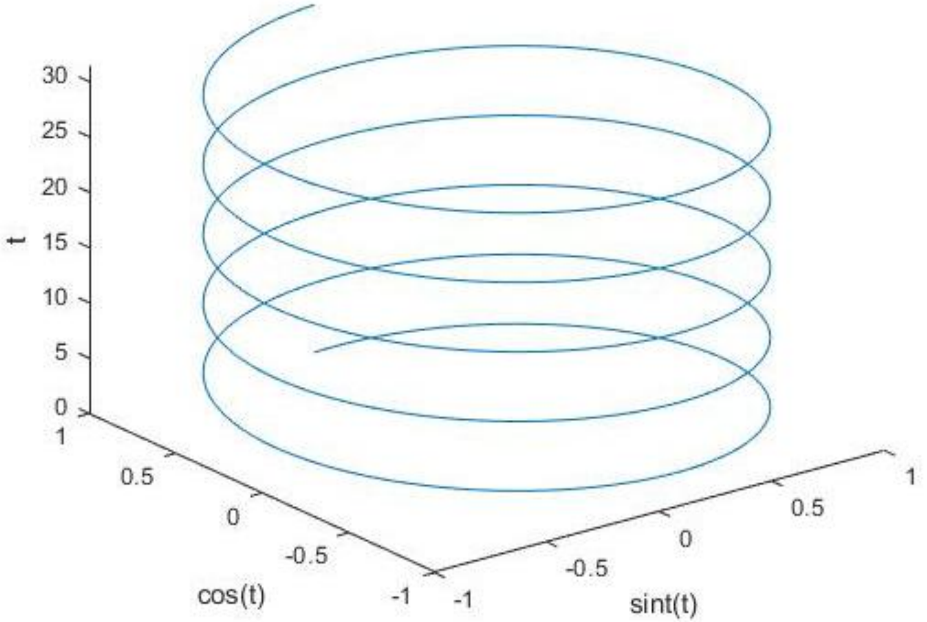
```
[x,y]= meshgrid(-2:0.2:2);
z=x.^2+y.^2;
figure
mesh(x,y,z);
figure
surf(x,y,z);
```





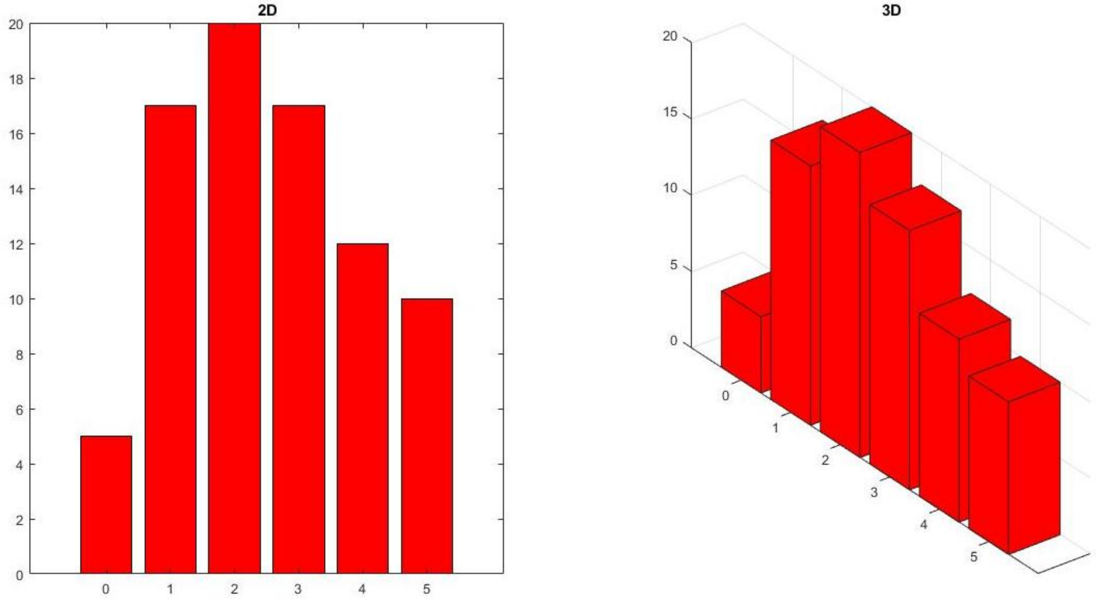
Plot 3-D Helix

```
t = 0:pi/50:10*pi;
st = sin(t);
ct = cos(t);
plot3(st,ct,t);
xlabel(‘sint(t)’);
ylabel(‘cos(t)’);
zlabel(‘t’);
```



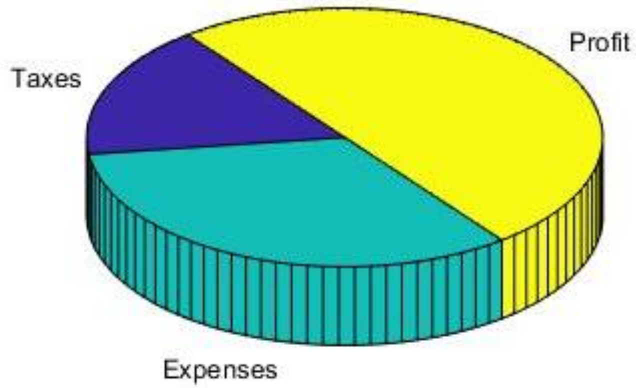
2-D/ 3-D bar graph of data

```
x= [0 1 2 3 4 5];
y=[5 17 20 17 12 10];
subplot(1,2,1);
bar(x, y, ‘r’); title(‘2D’);
subplot(1,2,1);
bar(x, y, ‘r’); title(‘3D’);
```



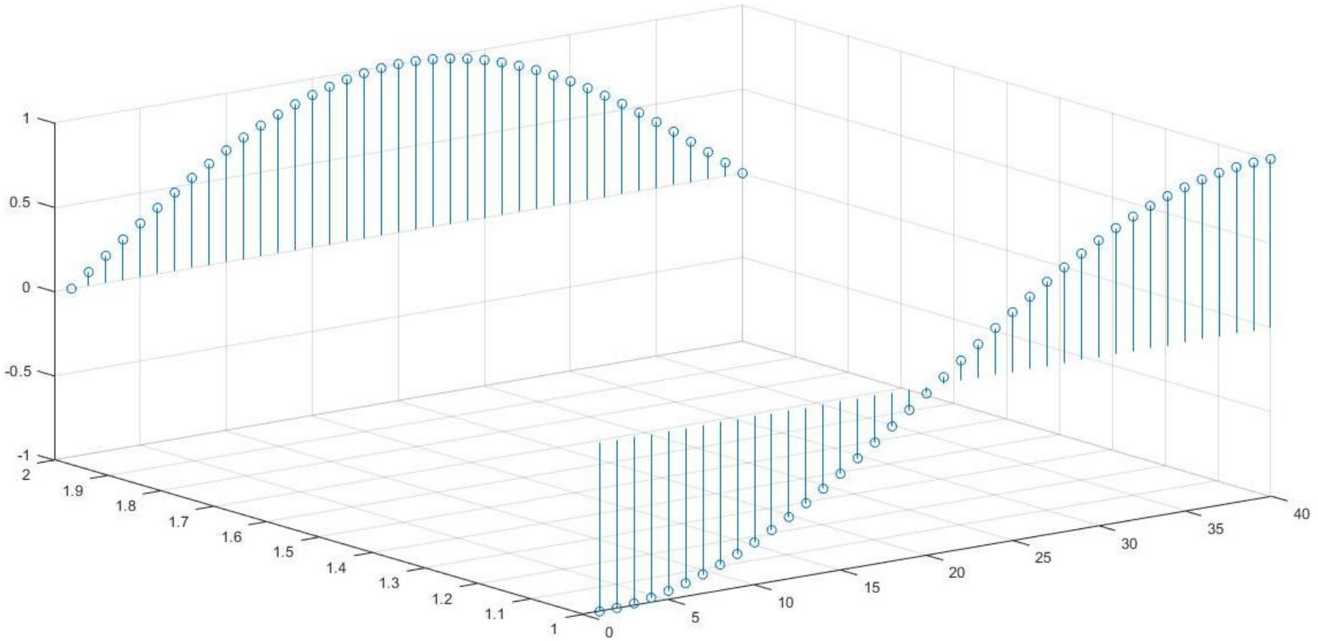
Create a 3-D pie chart and specify the text labels

```
x = 1:3;
labels = {'Taxes’, 'Expenses’, 'Profit’};
figure
pie3(x, labels);
```



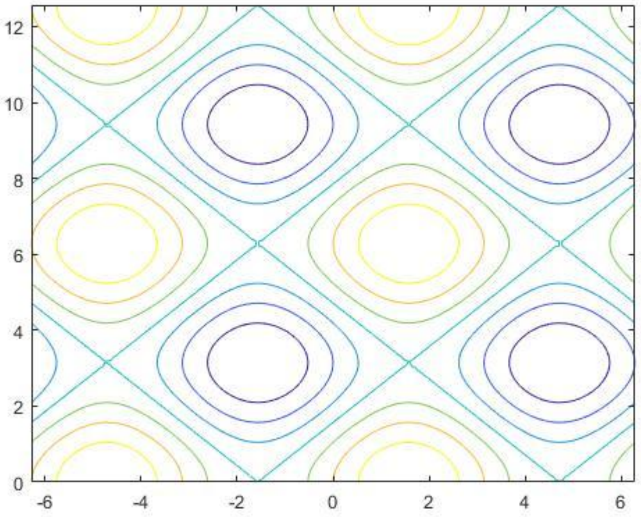
3-D stem plot of sine and cosine values between −π/2 and π/2 with a matrix input

```
figure
X = linspace(-pi/2,pi/2,40);
Z = [sin(X); cos(X)];
stem3(Z)
```



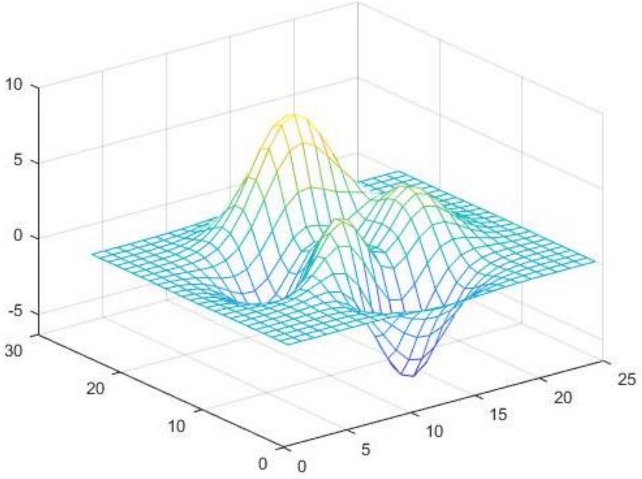
Contours of a Function

```
x = linspace(-2*pi,2*pi);
 y = linspace(0,4*pi);
[X,Y] = meshgrid(x,y);
 Z = sin(X)+cos(Y);
contour(X,Y,Z)
```



Mesh Plot - The mesh function creates a wireframe mesh. By default, the color of the mesh is proportional to the surface height

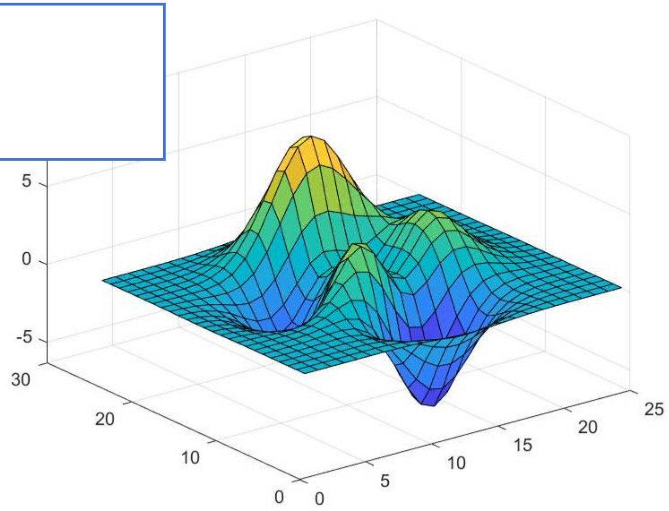
```
z = peaks(25);
 figure
mesh(z)
```



Surface Plot

The **surf** function is used to create a **3-D surface** plot.

```
figure
surf(z)
```



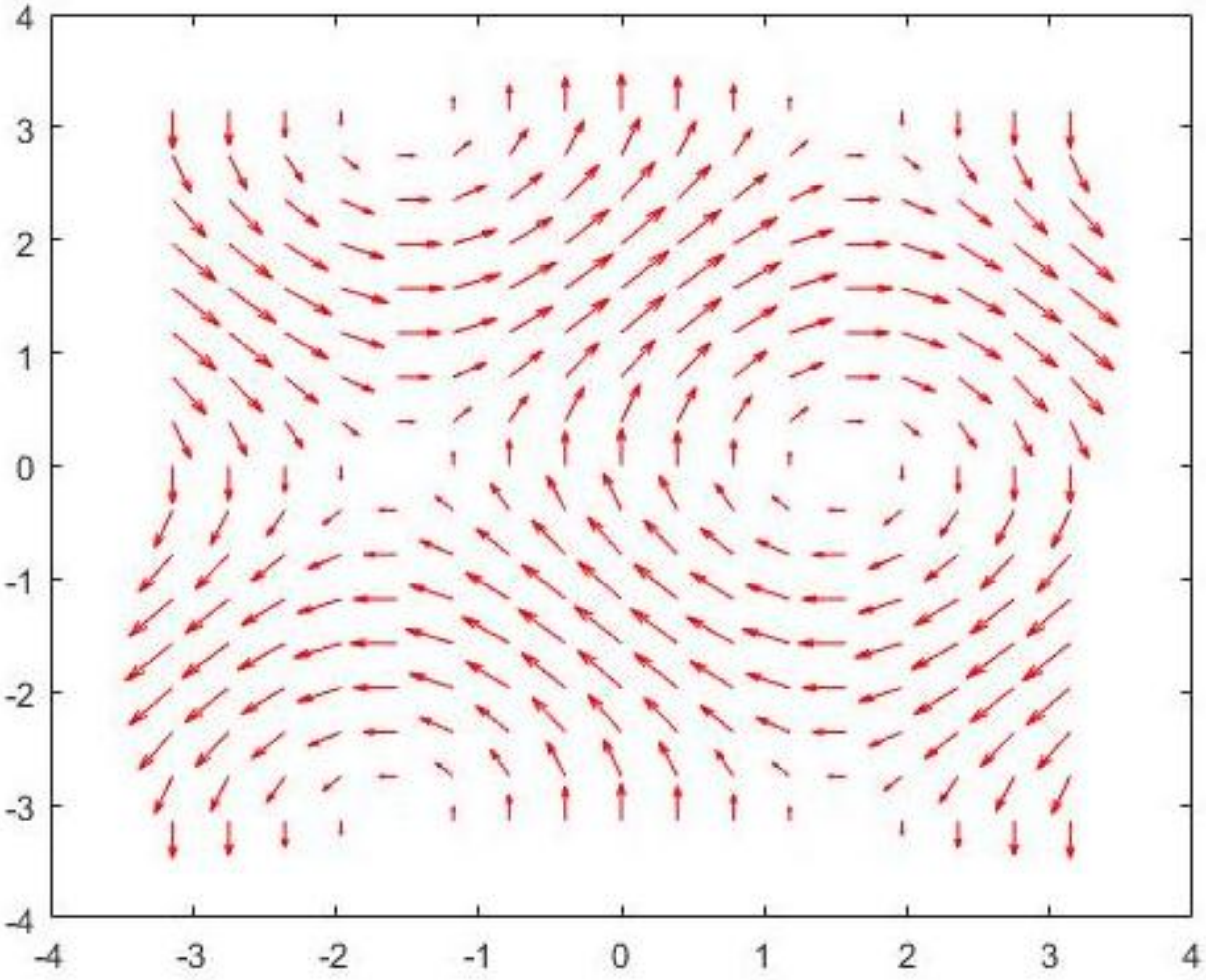
Quiver Plot

- The quiver function plots 2-D vectors as arrows

```
[X,Y] = meshgrid(-pi:pi/8:pi,-pi:pi/8:pi);
U = sin(Y);
V = cos(X);
quiver(X,Y,U,V,'r’);
```

(x_pos, y_pos, x_dir, y_dir, color)

x_pos and **y_pos** are the starting positions of the arrow while **x_dir** and **y_dir** are the directions of the arrow.

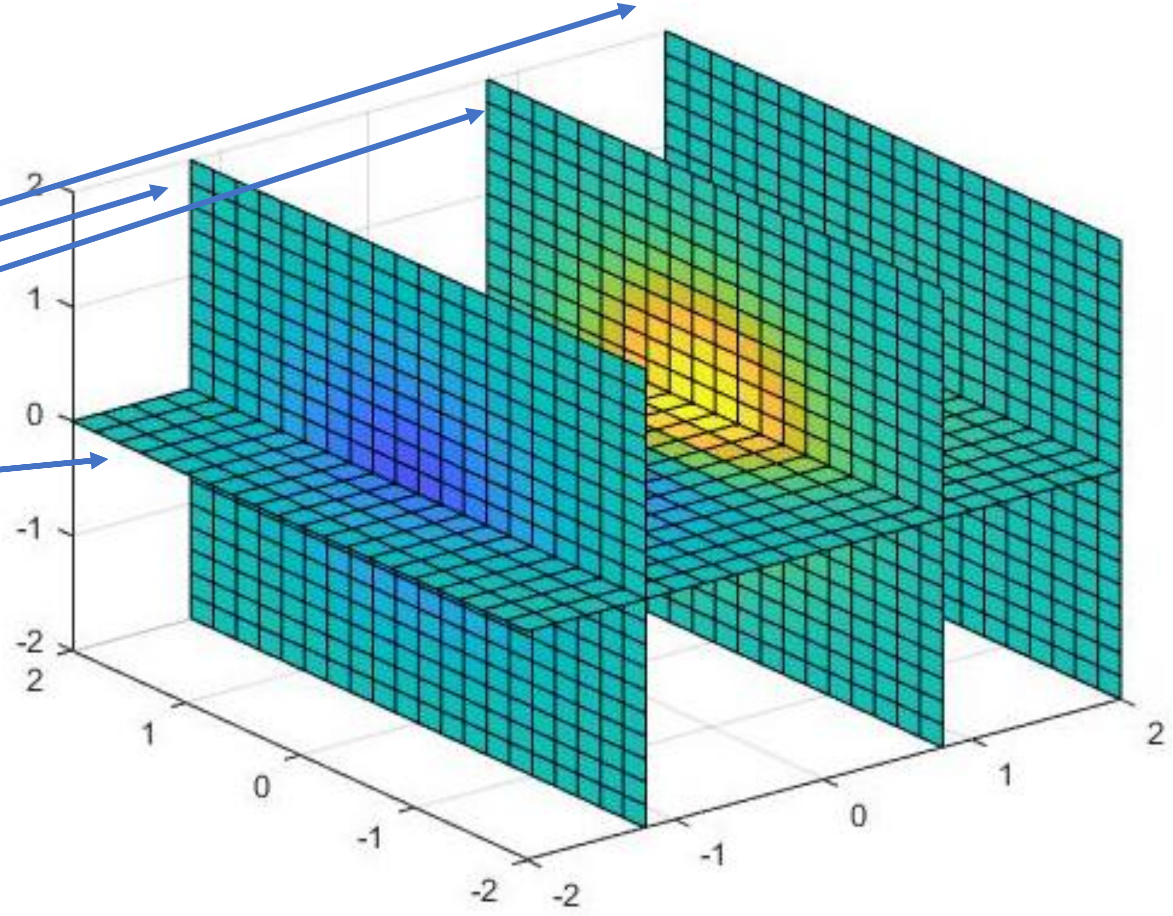


Slices through 3-D Volumes

- The slice function displays data at planes that slice through volumetric data

```
[X,Y,Z] = meshgrid(-2:.2:2);
V = X.*exp(-X.^2-Y.^2-Z.^2);

xslice = [-1.2,0.8,2];
yslice = [];
zslice = 0;
slice(X,Y,Z,V,xslice,yslice,zslice)
```

```
xslice  % location of y-z planes
yslice  % location of x-z plane
zslice % location of x-y planes
```

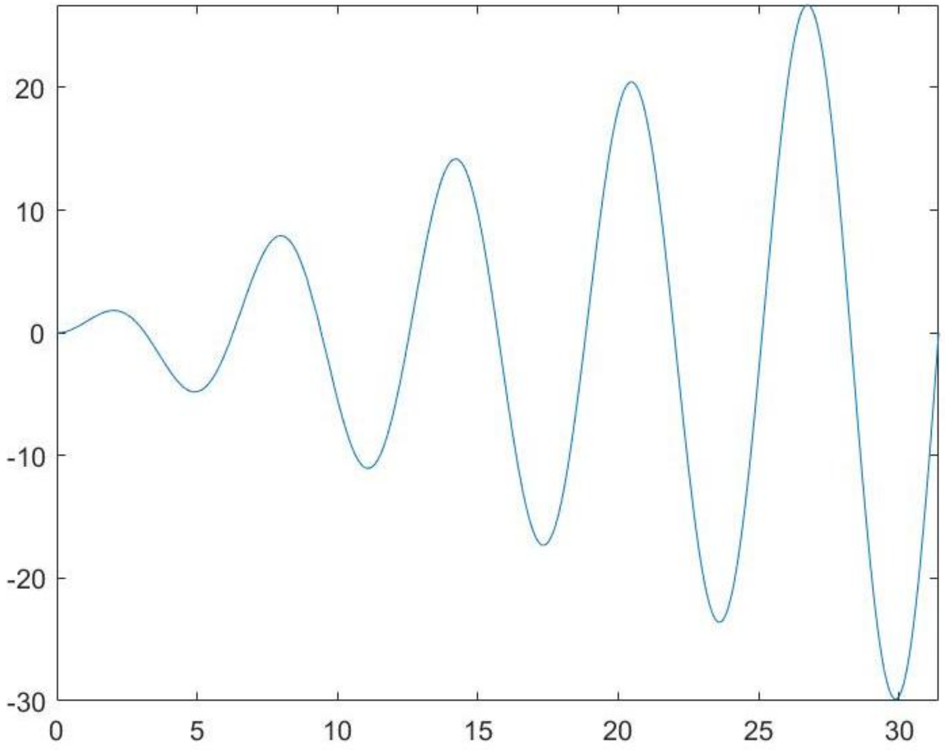


Plot expression or function - **fplot**

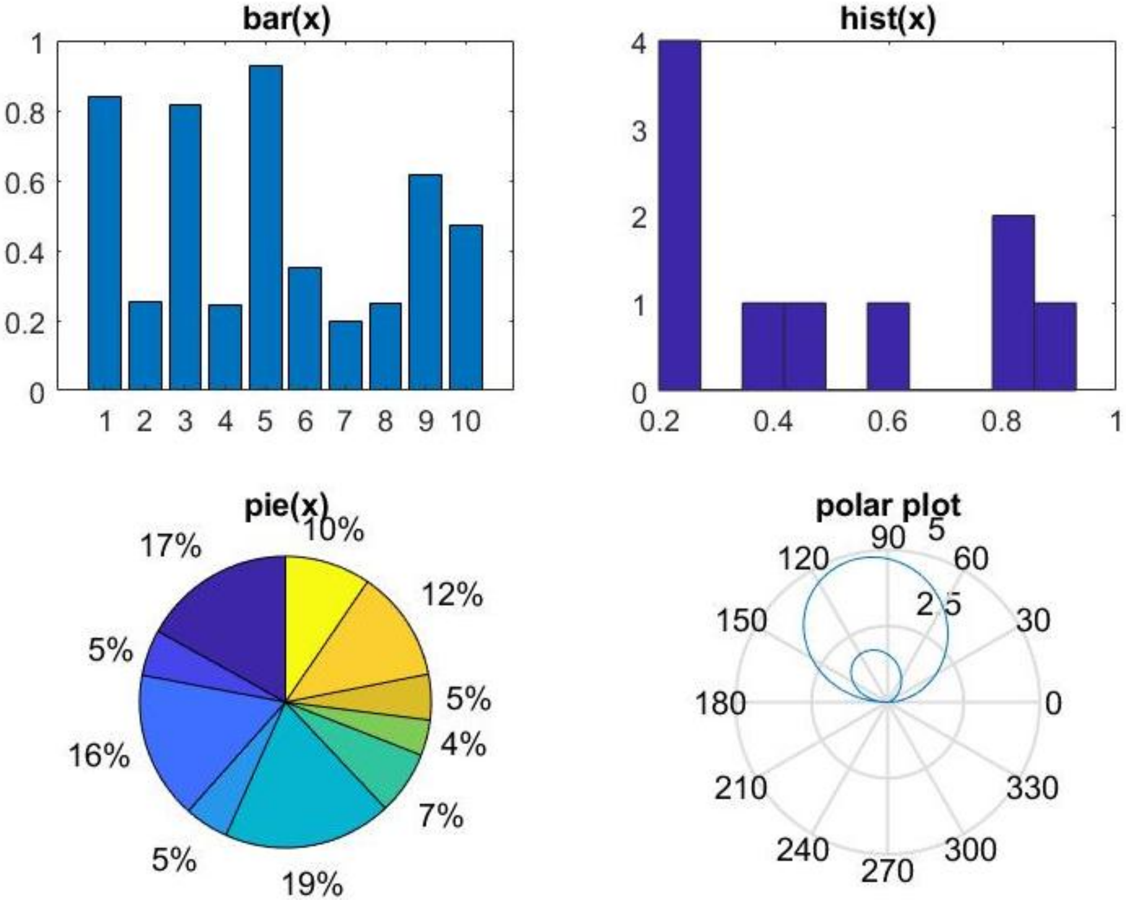
- Plot f(t)= tsint , 0 ≤ t ≤ T

- >> fplot(**@(x) x.*sin(x)**, [0 10*pi])

- fplot(f(x), **xinterval**) plots the curve
- defined by the function f(x)
- over the interval [xmin xmax] for x.



```
x=rand(10,1);

subplot(2,2,1);
bar(x);
title('bar(x)’);

subplot(2,2,2);
hist(x);%group of data points organised within specified ranges
title('hist(x)’);

subplot(2,2,3);
pie(x);
title('pie(x)’);

subplot(2,2,4);
t= linspace(0, 2*pi, 200);
polar(t, t.*sin(t)); % polar plots magnitude and phase
title('polar plot');
```


bar(x)
hist(x)
pie(x)
polar plot