

INTRODUCTION TO MATLAB GRAPHS

ECE 321 – PRINCIPLES OF COMMUNICATION

Tuesday, 09 December 2025

PLOTTING A BASIC LINE GRAPH

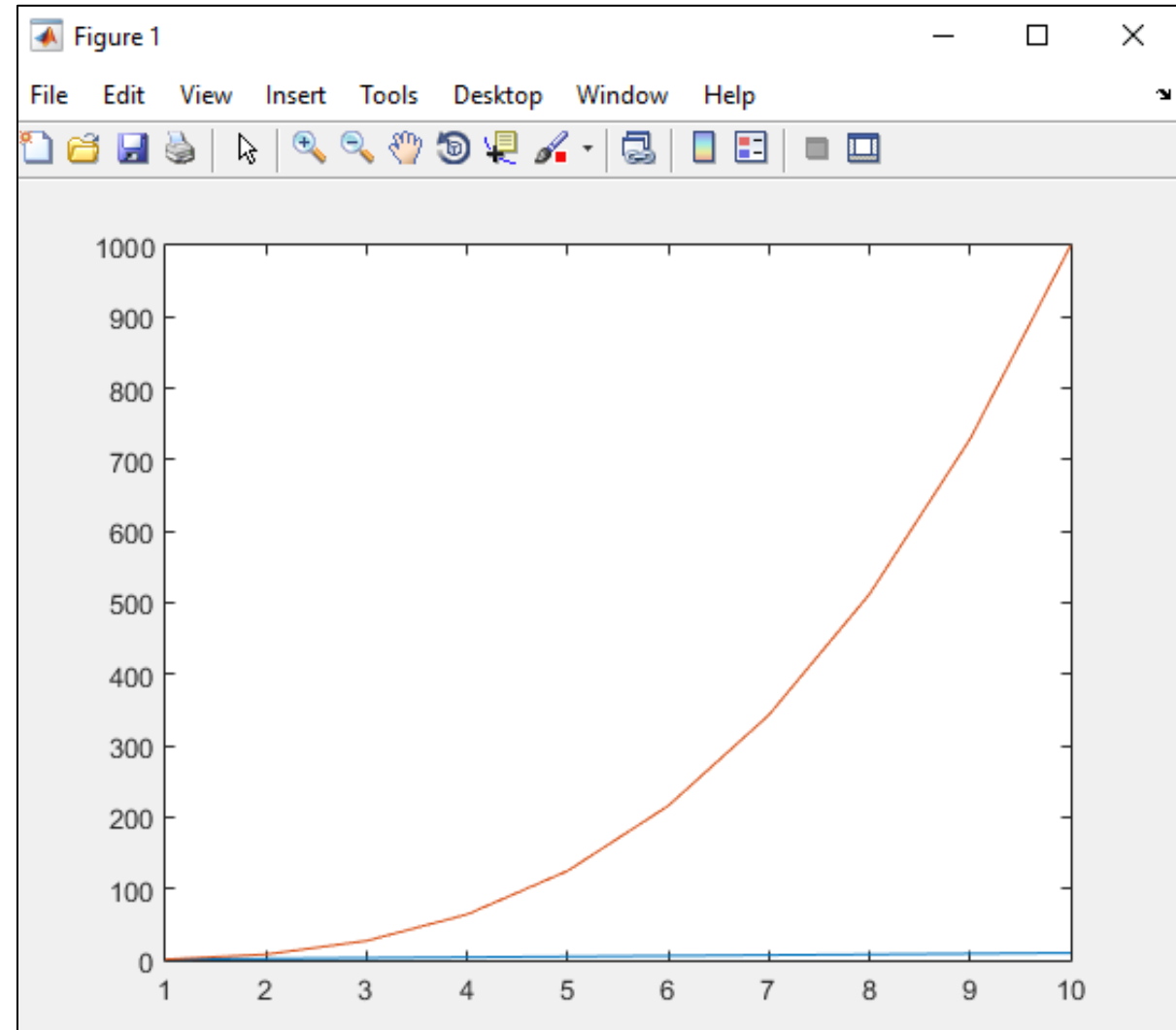
To display a line graph of function $y = x^3$

Enter

```
x = [1 2 3 4 5 6 7 8 9 10];
```

```
y = x.^3;
```

```
plot(x,y)
```



PLOTTING A CIRCLE

Objectives:

- To plot a circle centred at the point (4,3) with a radius equal to 2.
- To use axis equal to use equal data units along each coordinate direction.

Objective

`r = 2;`

`xc = 4;`

`yc = 3;`

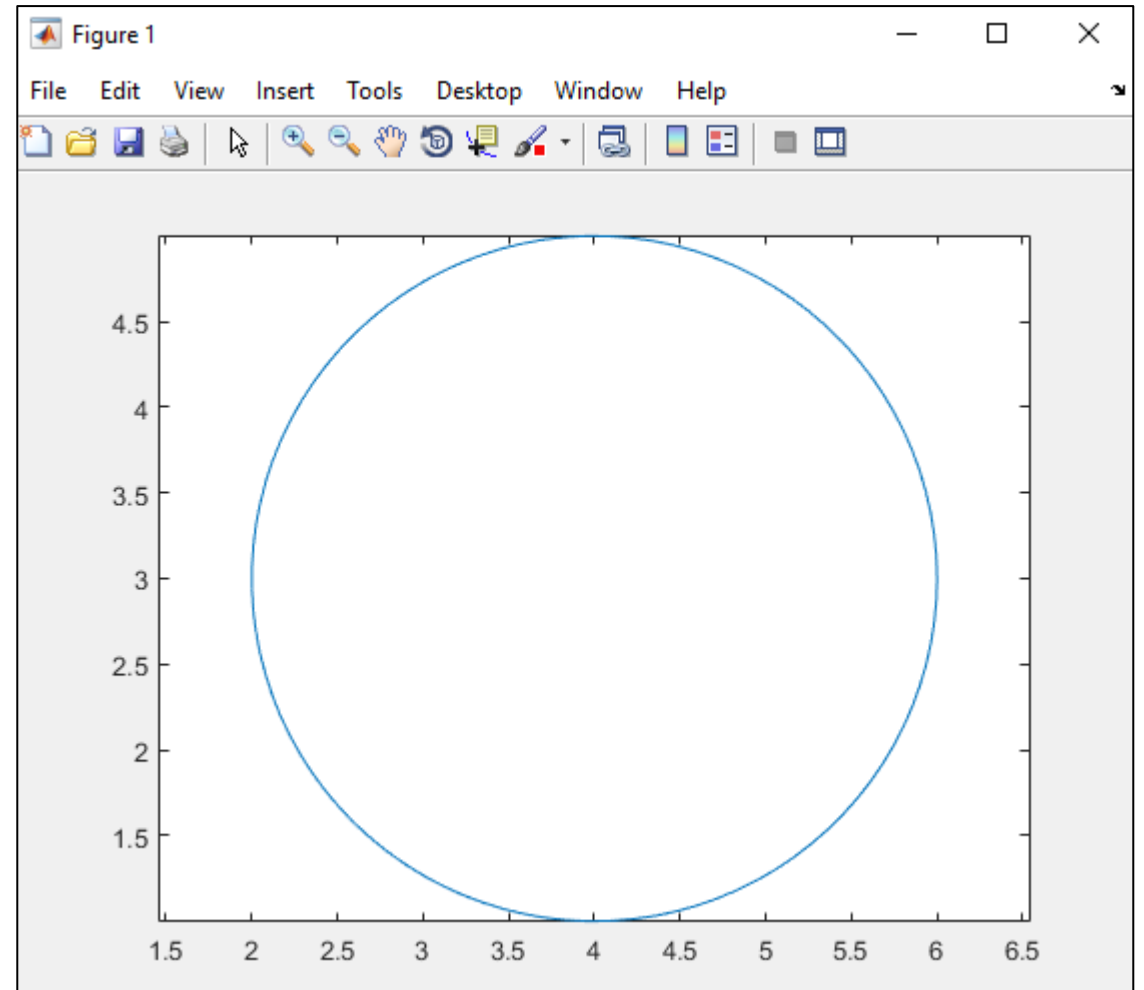
`theta = linspace(0,2*pi);`

`x = r*cos(theta) + xc;`

`y = r*sin(theta) + yc;`

`plot(x,y)`

`axis equal`



PLOTTING SINGLE LINE GRAPH OF FUNCTION: $y = \sin\left(\frac{2\pi n}{100}\right)$

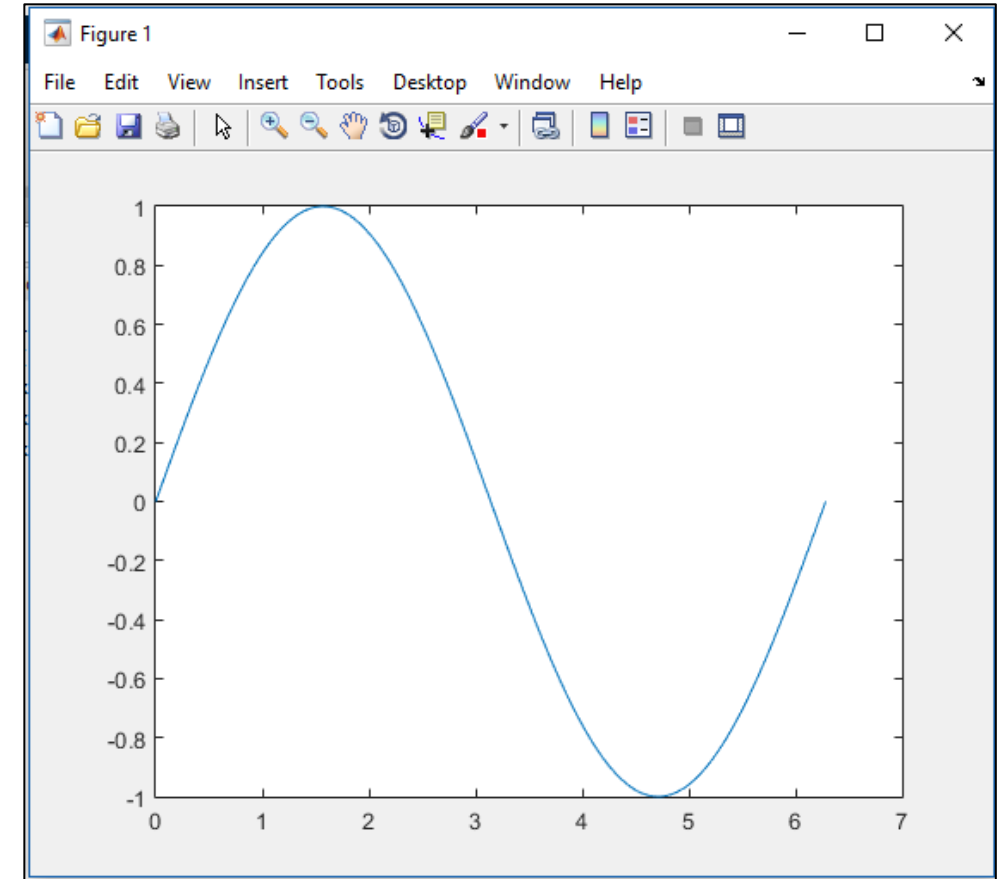
Define x as a vector of linearly spaced values between 0 and 2π using increment of $\frac{\pi}{100}$.

Enter

$x = 0 : \pi/100 : 2*\pi;$

$y = \sin(x);$

$\text{Plot}(x,y)$



PLOTTING MULTIPLE LINE GRAPH OF FUNCTIONS:

$$y1 = \sin\left(\frac{2\pi n}{100}\right) \text{ and } y2 = \cos\left(\frac{2\pi n}{100}\right)$$

- Define x as 100 linearly spaced values between -2π and 2π .
- Define y1 and y2 as sine and cosine values of x.
- Create a line plot of both sets of data.

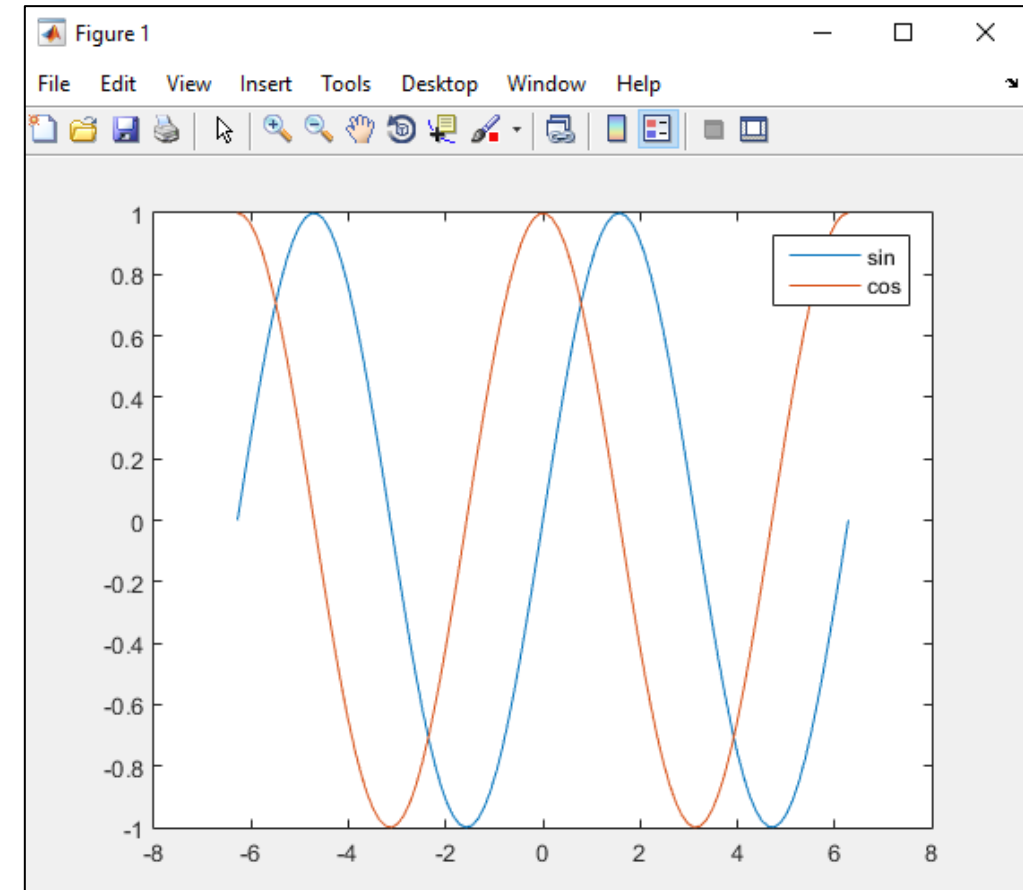
Enter

```
x = linspace(-2*pi,2*pi);
```

```
y1 = sin(x);
```

```
y2 = cos(x);
```

```
plot(x,y1,x,y2)
```



PLOTTING A LINE GRAPH FROM A MATRIX

Define y as the 8-by-8 matrix returned by the magic function and plot.

Enter

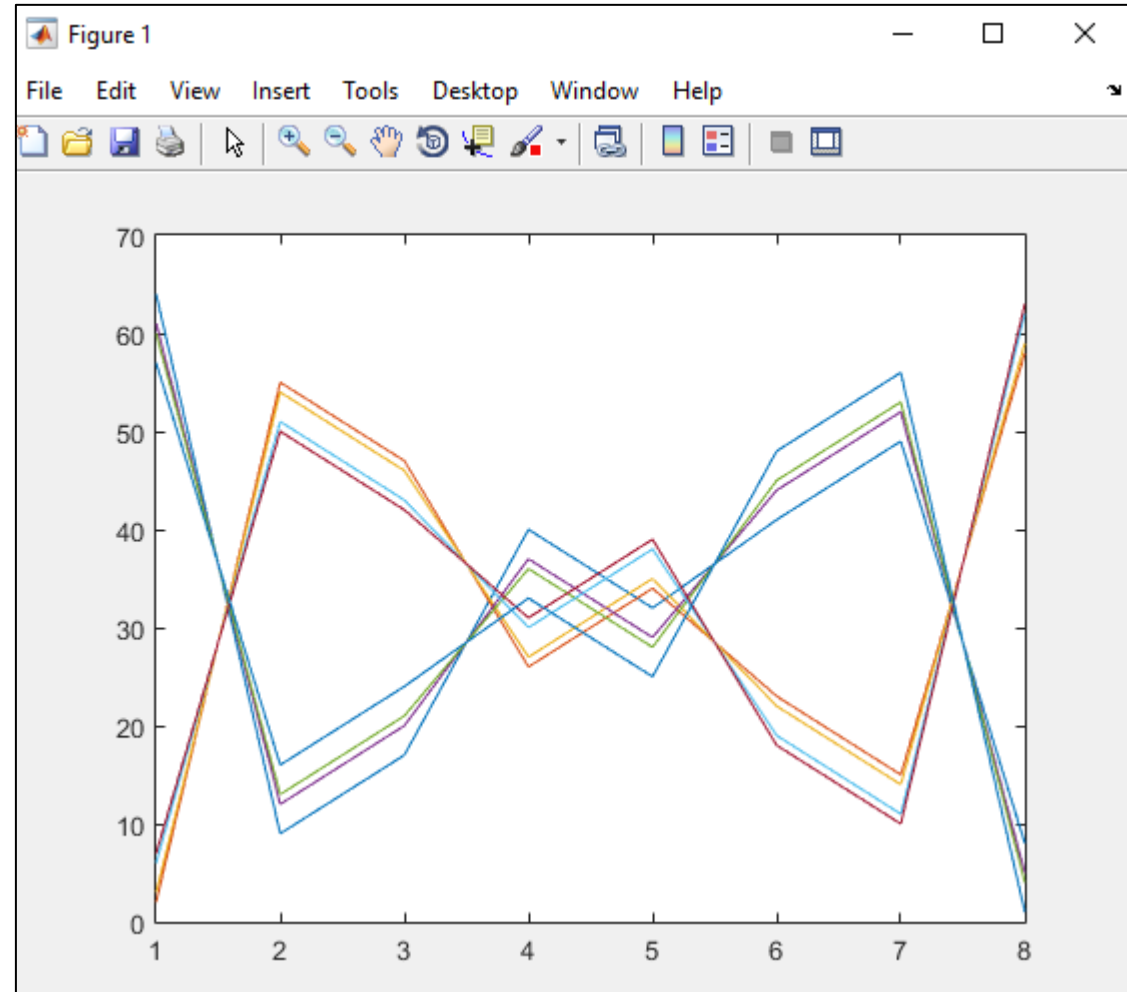
$Y = \text{magic}(8)$

```
>> Y = magic(8)
```

Y =

64	2	3	61	60	6	7	57
9	55	54	12	13	51	50	16
17	47	46	20	21	43	42	24
40	26	27	37	36	30	31	33
32	34	35	29	28	38	39	25
41	23	22	44	45	19	18	48
49	15	14	52	53	11	10	56
8	58	59	5	4	62	63	1

```
>> plot(Y)
```



PLOTTING USING LINE STYLES

- Plot three sine curves with a small phase shift between each line.
- Use the default line style for the first line.
- Specify a dashed line style for the second line
- Specify a dotted line style for the third line.

Enter

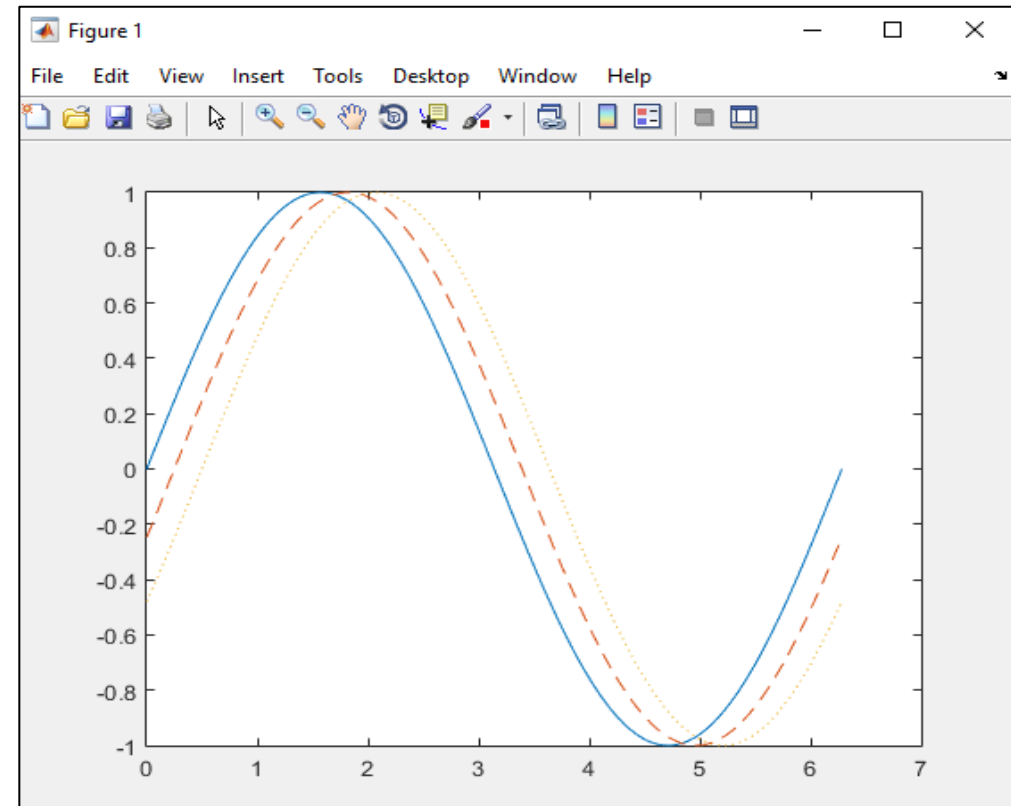
```
x = 0:pi/100:2*pi;
```

```
y1 = sin(x);
```

```
y2 = sin(x-0.25);
```

```
y3 = sin(x-0.5);
```

```
plot(x,y1,x,y2,'--',x,y3,':')
```



PLOTTING USING LINE STYLE, COLOUR & MARKER

- Plot three sine curves with a small phase shift between each signal.
- Use a red line with no markers for the first sine curve.
- Use a blue dashed line with circle markers for the second sine curve.
- Use only cyan star markers for the third sine curve.

ENTER

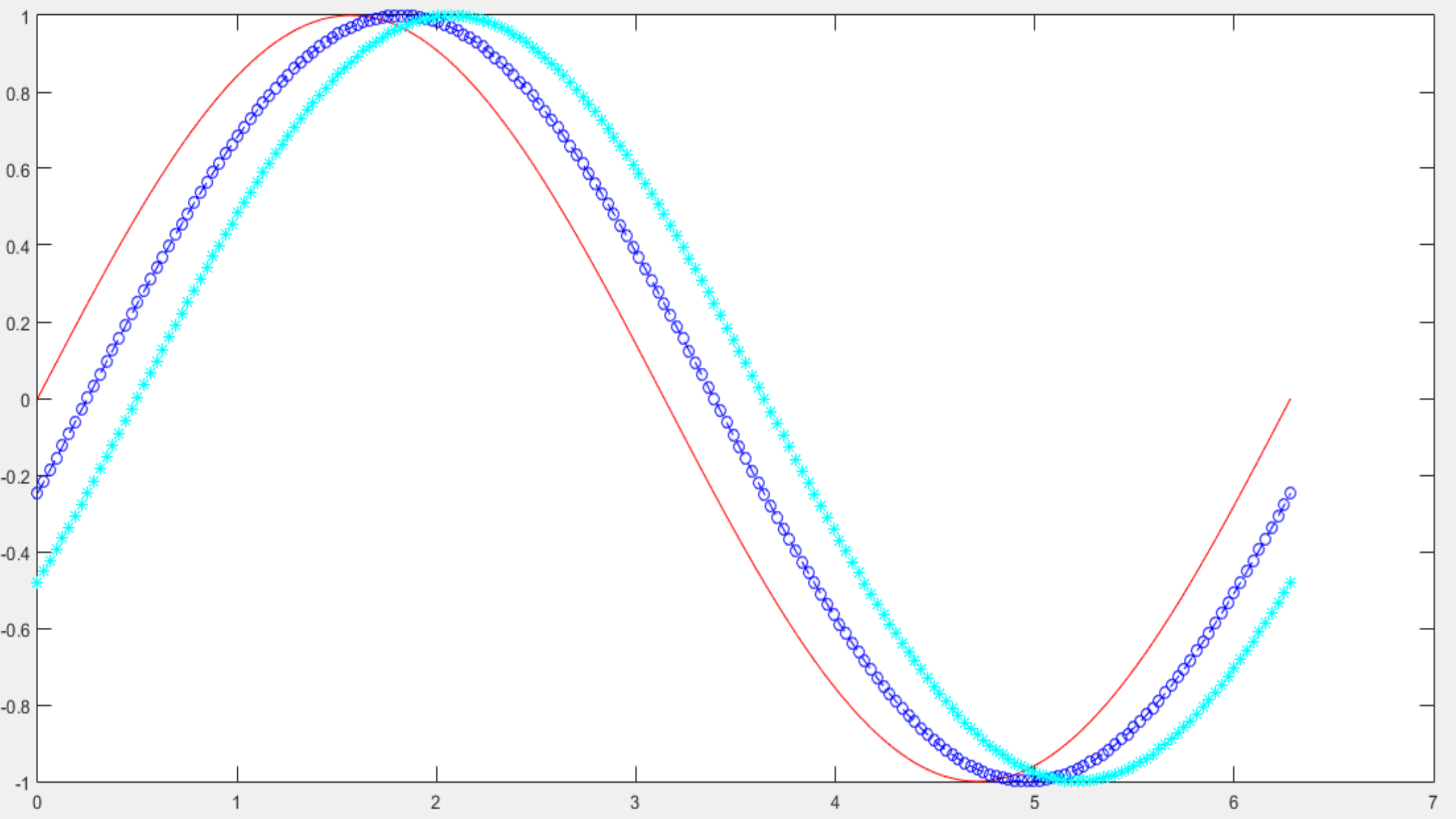
```
x = 0 : pi/10 : 2*pi;
```

```
y1 = sin(x);
```

```
y2 = sin(x-0.25);
```

```
y3 = sin(x-0.5);
```

```
plot(x,y1,'r',x,y2,'b--o',x,y3,'c*')
```



ADDING TITLE AND AXIS LABELS

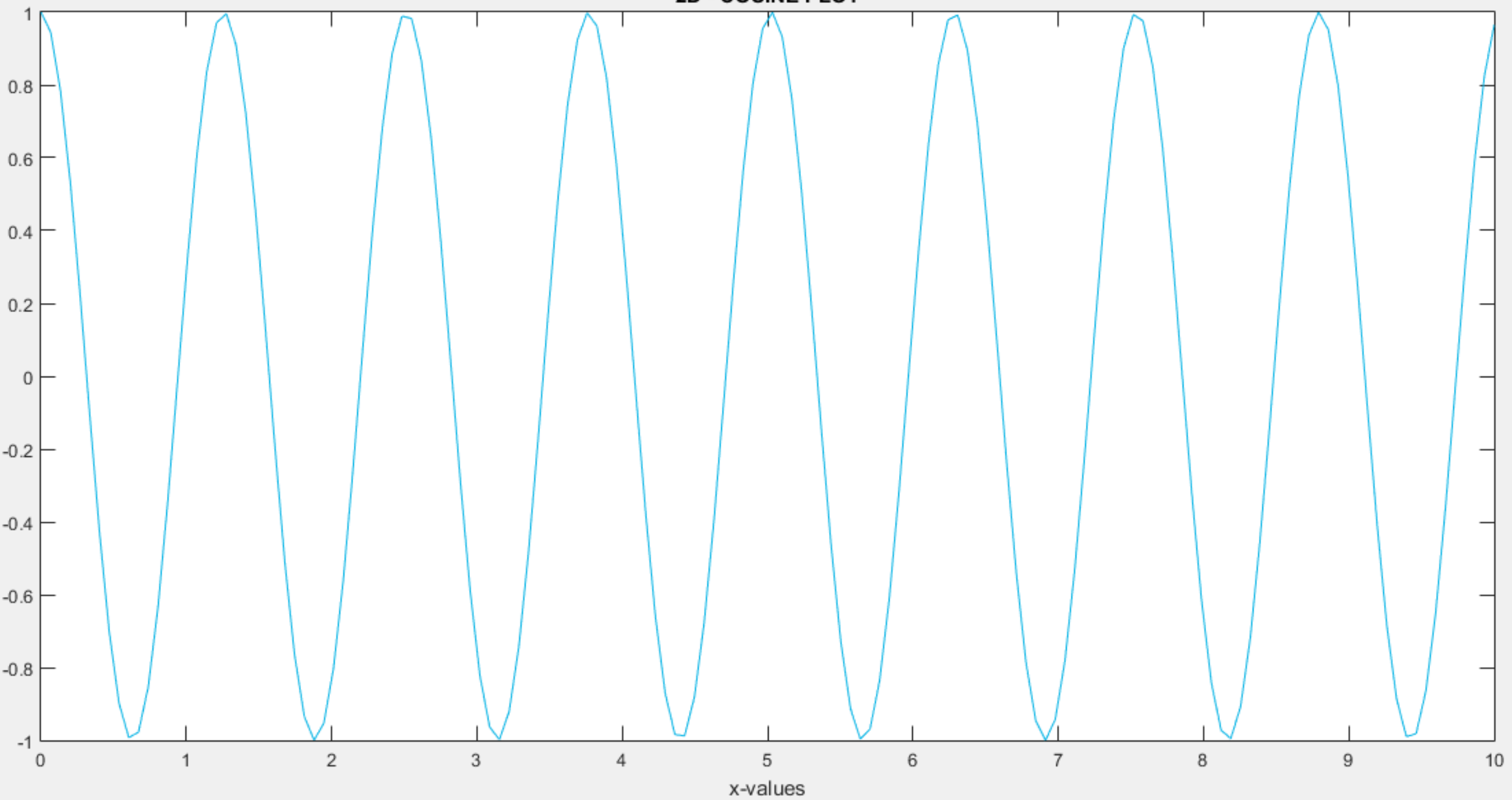
Objective:

- Use the linspace function to define x as a vector of 150 values between 0 and 10. Define y as cosine values of x.
- Create a 2-D line plot of the cosine curve.
- Change the line color to a shade of blue-green using an RGB color value.
- Add a title and axis labels to the graph using the title, xlabel, and ylabel functions.

ENTER:

```
x = linspace(0,10,150);  
y = cos(5*x);  
plot(x,y,'Color',[0,0.7,0.9]);  
title('2-D Line Plot');  
xlabel('x-values');  
ylabel('cos(5x)')
```

2D - COSINE PLOT



PLOTTING 3D GRAPHS

- Define **x** as values between **0** and **10π** .
- Define **st** and **ct** as vectors of sine and cosine values.
- Use **plot3** function to display graph.
- Label axes; insert title; and turn on grid.

ENTER

```
t = 0:pi/50:10*pi;
```

```
st = sin(t);
```

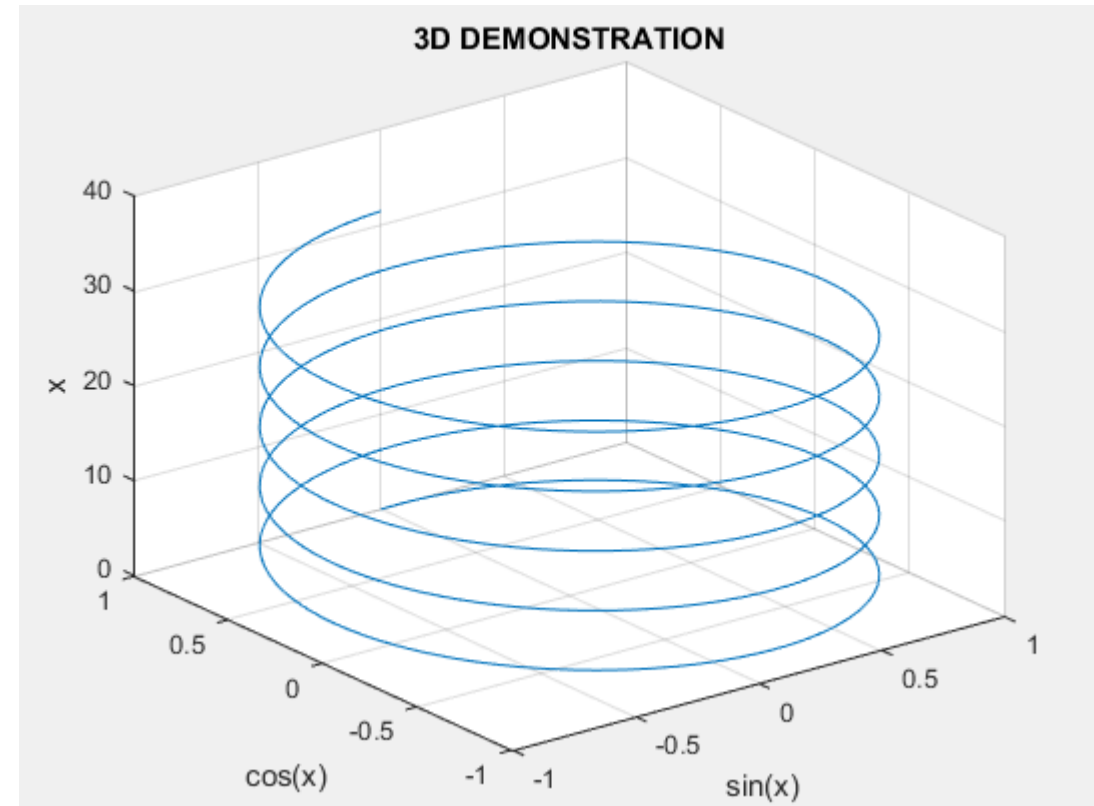
```
ct = cos(t);
```

```
plot3(st, ct, t);
```

```
xlabel('sin(t)');
```

```
ylabel('cos(t)');
```

```
title('3D DEMONSTRATION')
```

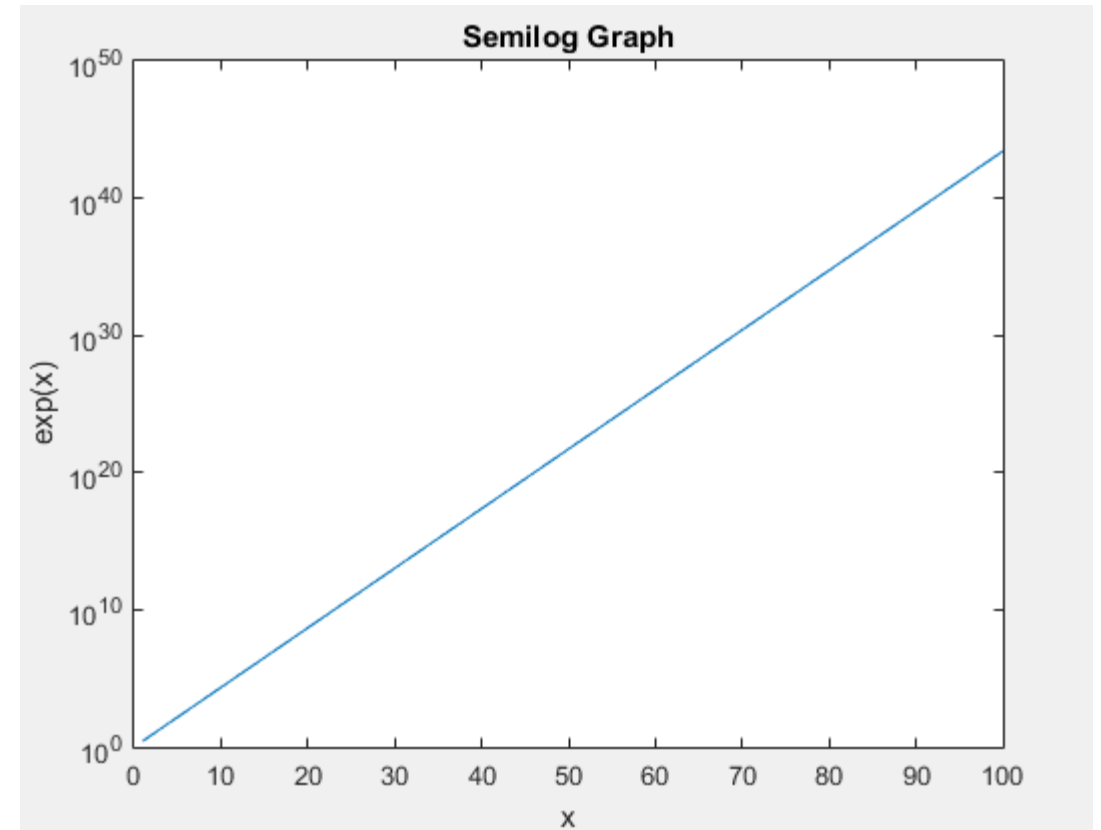


PLOTTING SEMILOG GRAPHS

Semilogy command is used to create a plot with a logarithmic scale for the y-axis and a linear scale for the x-axis.

ENTER

```
x = 1:100;  
y = exp(x);  
Semilogy(x,y);  
xlabel('x');  
ylabel('exp(x)');  
title('Semilog Graph')
```



PLOTTING LOG-LOG GRAPH

Used to create a plot with a logarithmic scale on both y-axis.

ENTER:

`x=1:100;`

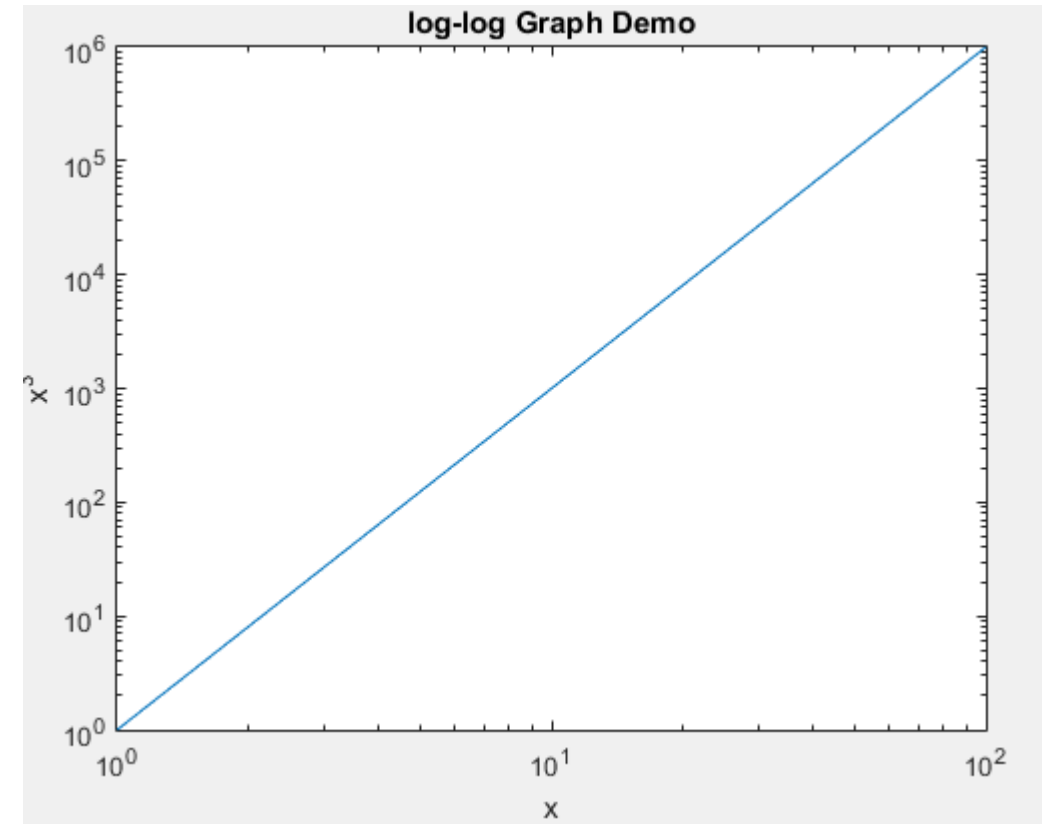
`y=x.^3;`

`loglog(x,y)`

`xlabel('x');`

`ylabel('x^3');`

`title('log-log Graph Demo')`



USING SURF FUNCTION

surf(X,Y,Z) plots the values in matrix Z as heights above a grid in the x-y plane defined by X and Y.

Z is used for the colour data, so colour is proportional to height.

ENTER

```
x = 1:0.5:10;
```

```
y = 1:20;
```

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

```
Z = sin(X) + cos(Y);
```

```
C = X.*Y;
```

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

```
colorbar
```

