MAT 1206 – Introduction to MATLAB

CHAPTER 02: Fundamental Operators and Commands

Lesson 4: Plotting and Graphics

Content

- Plotting
  - Linear plots
  - line types, plot symbols and colors
  - Adding title, label, grid lines and scaling
  - Multiple functions on the same graph
- Graphics
  - Bar charts
  - Pie chart
  - Contours
  - Three-dimensional plots

Linear plots

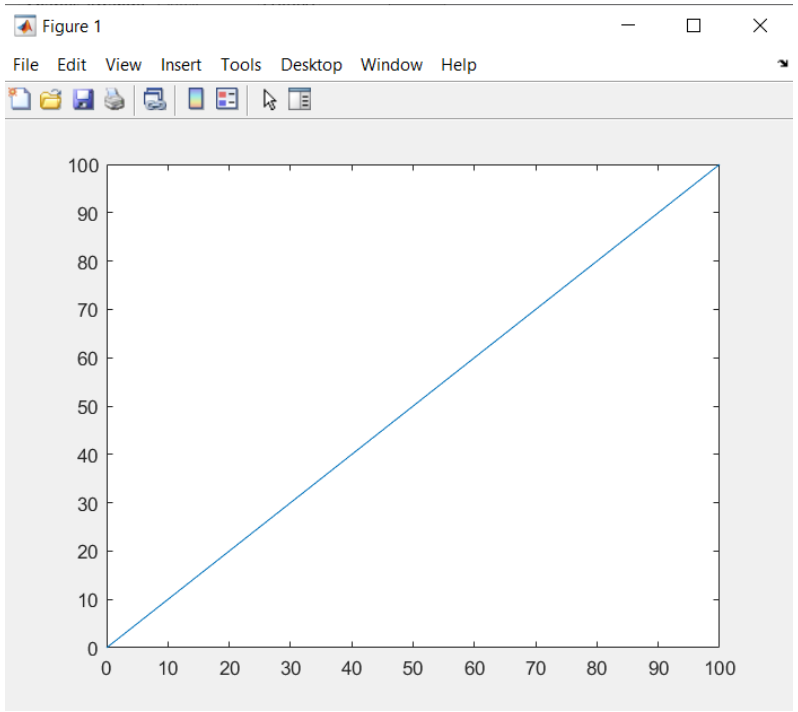
To plot the graph of a function, you need to take the following steps:

- Define **x**, by specifying the range of values for the variable x, for which the function is to be plotted
- Define the function, **y = f(x)**
- Call the plot command, as **plot(x, y)**

Cont.

Following example plot the simple function y = x for the range of values for x from 0 to 100, with an increment of 5.

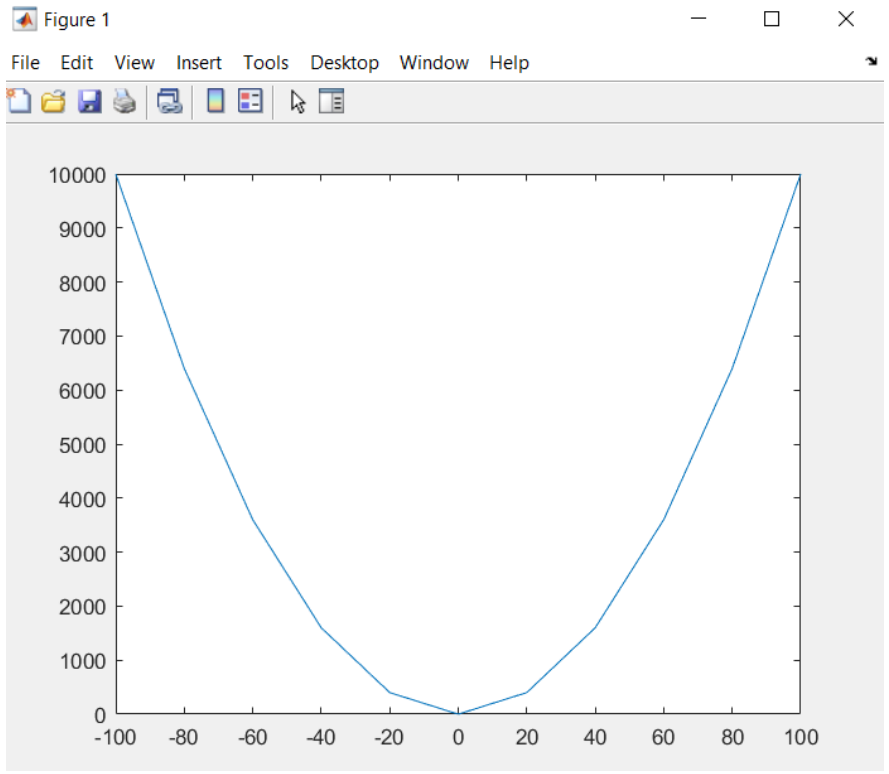
```
x = [0:5:100];
y = x;
plot(x, y)
```



Cont.

Let us take one more example to plot the function $y = x^2$.

```
x = [-100:20:100];
y = x.^2;
plot(x, y)
```



Reduce the increment to 5:

```
x = [-100:5:100];
y = x.^2;
plot(x, y)
```

line types, plot symbols and colors

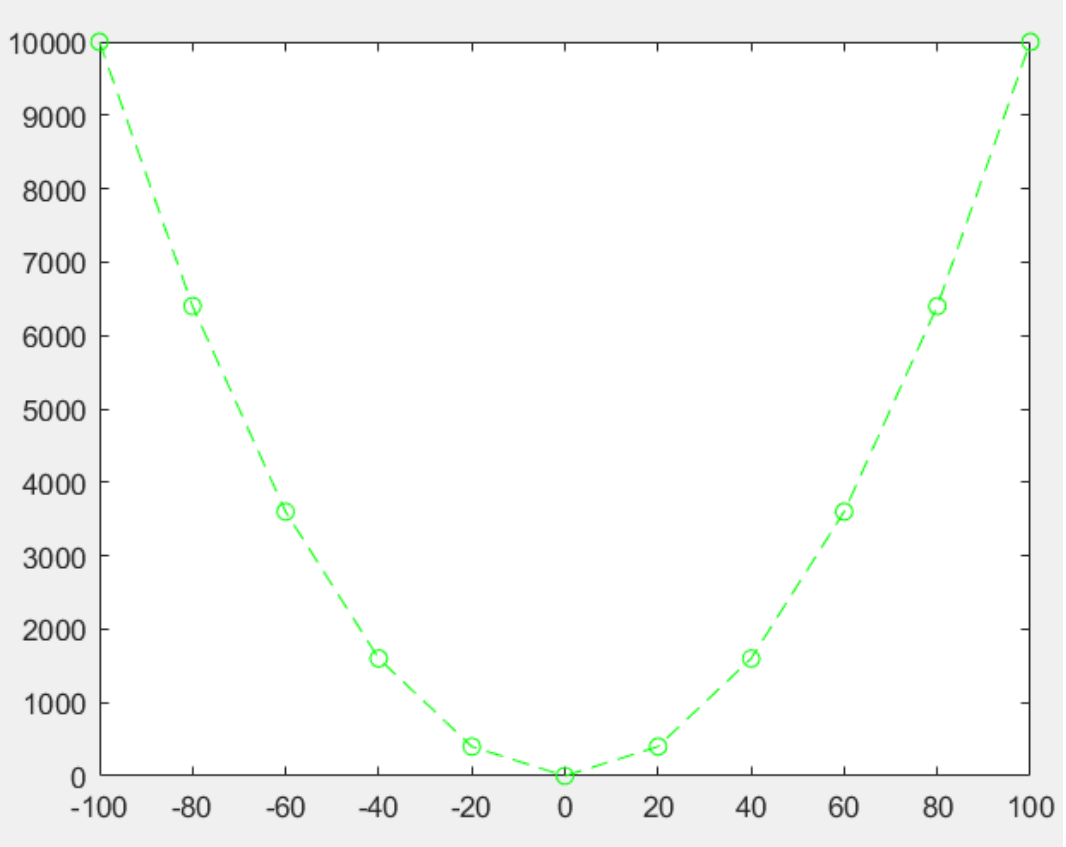
Various line types, plot symbols and colors may be obtained with plot(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Cont.

Example:

```
x = [-100:5:100];
y = x.^2;
plot(x, y,'go--')
```



Adding title, label, grid lines and scaling

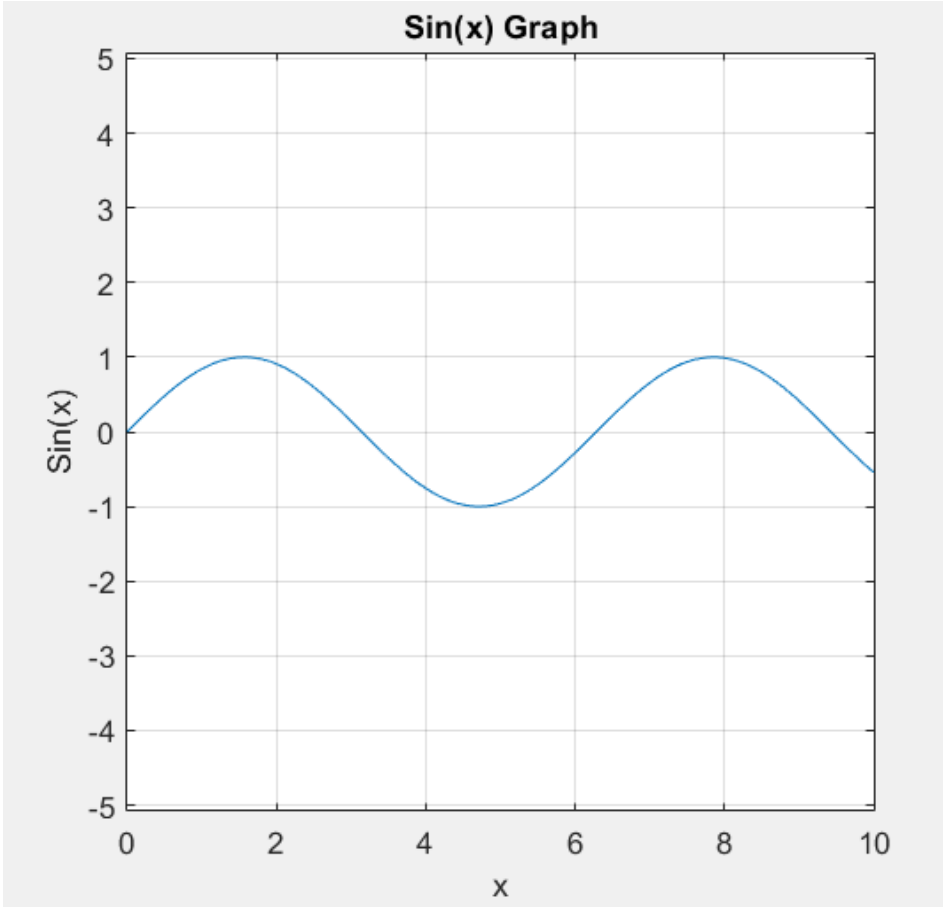
MATLAB allows to add title, labels along the x-axis and y-axis, grid lines and also to adjust the axes to spruce up the graph.

- The **xlabel** and **ylabel** commands generate labels along x-axis and y-axis.
- The **title** command allows you to put a title on the graph.
- The **grid** on command allows you to put the grid lines on the graph.
- The **axis equal** command allows generating the plot with the same scale factors and the spaces on both axes.
- The **axis square** command generates a square plot.

Cont.

Example:

```
 x = [0:0.01:10];
y = sin(x);
plot(x, y), xlabel('x'),
ylabel('Sin(x)'), title('Sin(x) Graph'),
grid on, axis equal
```

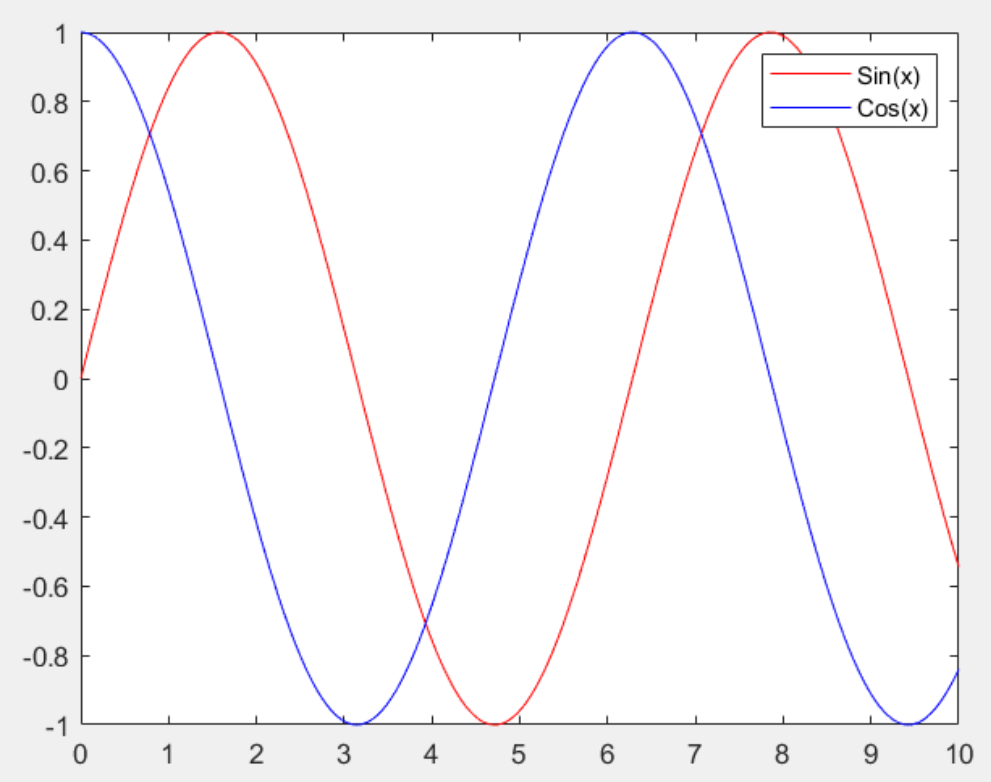


Multiple functions on the same graph

You can draw multiple graphs on the same plot.

Example:

```
 x = [0 : 0.01: 10];
y = sin(x);
g = cos(x);
plot(x, y,'r', x, g, 'b'),
legend('Sin(x)', 'Cos(x)')
```



Cont.

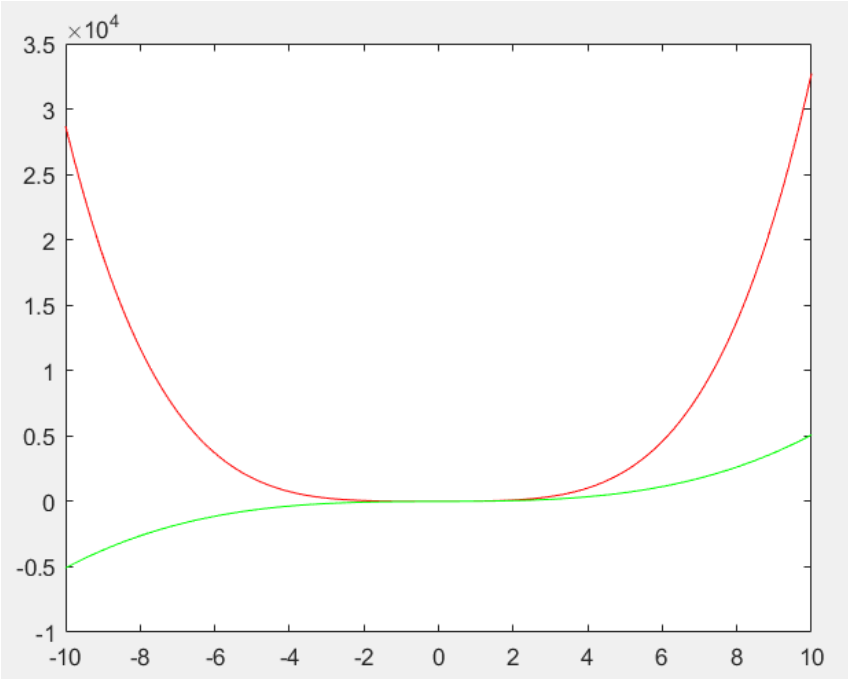
Example

Let us draw the graph of two polynomials

$f(x) = 3x^4 + 2x^3 + 7x^2 + 2x + 9$ and

$g(x) = 5x^3 + 9x + 2$

```
x = [-10 : 0.01: 10];
y = 3*x.^4 + 2 * x.^3 + 7 * x.^2 + 2 * x + 9;
g = 5 * x.^3 + 9 * x + 2;
plot(x, y, 'r', x, g, 'g')
```



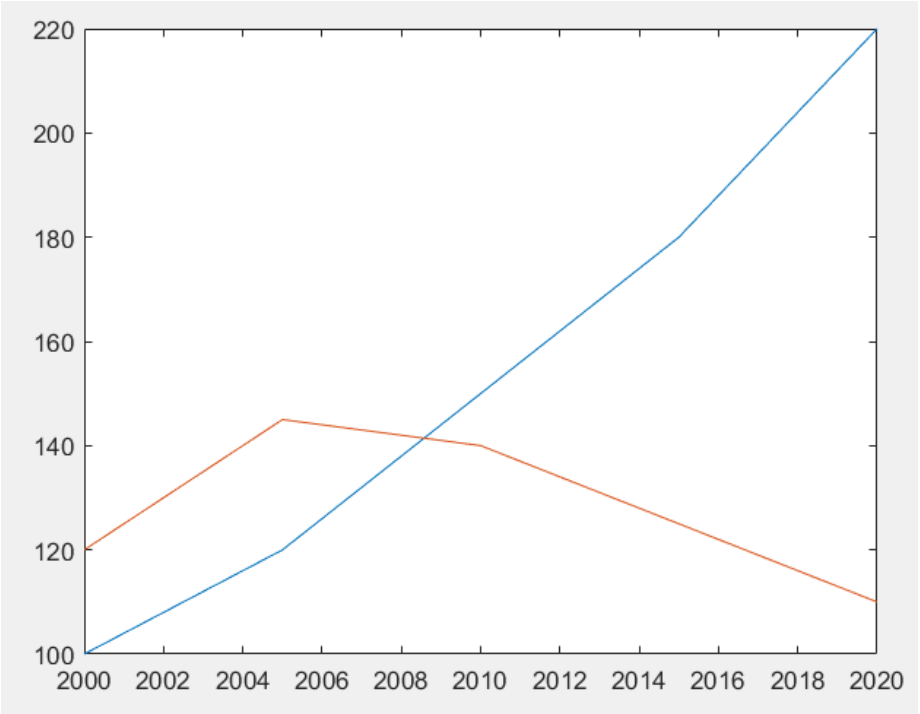
Cont.

Example

Sales of two products over the years

```
years=[2000:5:2020];
Sales_of_P1=[100 120 150 180 220];
Sales_of_P2=[120 145 140 125 110];

plot(years,Sales_of_P1,years,Sales_of_P2)
```



Graphics

- Bar Charts
- Pie Charts
- Contours
- Three-dimensional Plots

Bar Charts

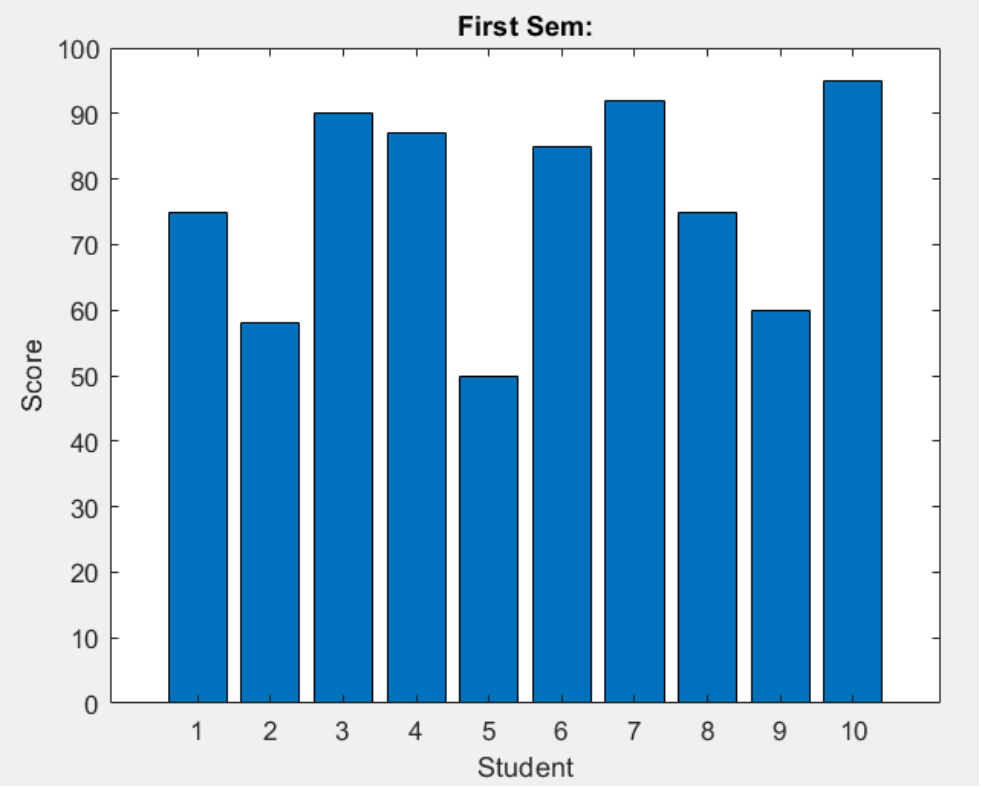
The **bar** command draws a two dimensional bar chart.

Example:

Let us have an imaginary classroom with 10 students. We know the percent of marks obtained by these students are 75, 58, 90, 87, 50, 85, 92, 75, 60 and 95. We will draw the bar chart for this data.

Cont.

```
x = [1:10];
y = [75, 58, 90, 87, 50, 85, 92, 75, 60, 95];
bar(x,y), xlabel('Student'),ylabel('Score'),
title('First Sem:')
```



Pie Chart

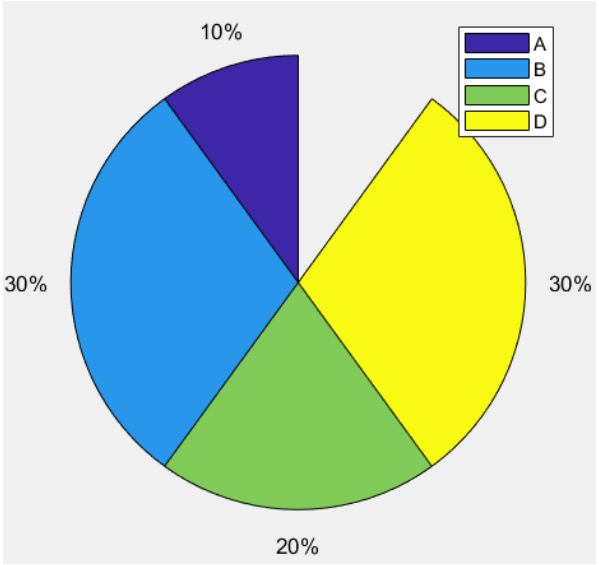
pie(X) draws a pie chart using the data in X. Each slice of the pie chart represents an element in X.

- If sum(X) ≤ 1, then the values in X directly specify the areas of the pie slices. pie draws only a partial pie if sum(X) < 1.

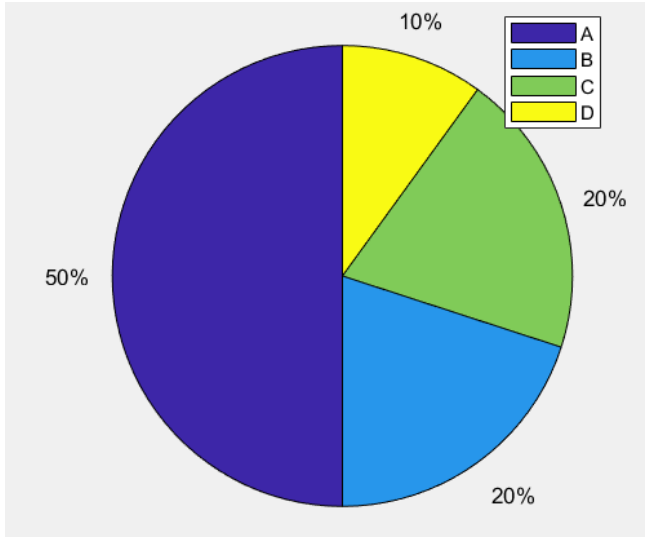
- If sum(X) > 1, then pie normalizes the values by X/sum(X) to determine the area of each slice of the pie.

Cont.

```
x=[0.1 0.3 0.2 0.3];
pie(x)
legend('A', 'B', 'C', 'D')
```




```
x=[50 20 20 10];
pie(x)
legend('A', 'B', 'C', 'D')
```



Contours

A contour line of a function of two variables is a curve along which the function has a constant value. Contour lines are used for creating contour maps by joining points of equal elevation above a given level, such as mean sea level.

MATLAB provides a contour function for drawing contour maps.

Example

Let us generate a contour map that shows the contour lines for a given function g = f(x, y). This function has two variables. So, we will have to generate two independent variables, i.e., two data sets x and y. This is done by calling the **meshgrid** command.

Cont.

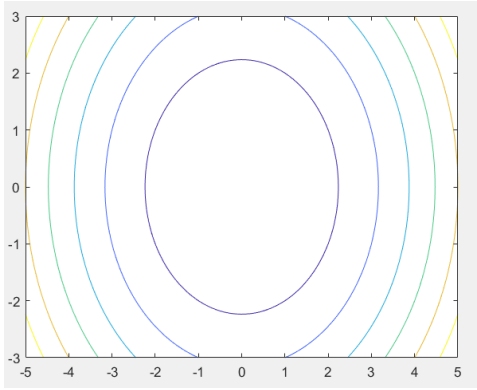
The **meshgrid** command is used for generating a matrix of elements that give the range over x and y along with the specification of increment in each case.

Let us plot our function g = f(x, y), where $-5 \leq x \leq 5$, $-3 \leq y \leq 3$. Let us take an increment of 0.1 for both the values. The variables are set as:

```
[x,y]=meshgrid(-5:0.1:5,-3:0.1:3);
```

Lastly, we need to assign the function. Let our function be: $x^2 + y^2$

```
f=x.^2+y.^2;
contour(x,y,f)  % call the contour function
```



Three-dimensional Plots

Three-dimensional plots basically display a surface defined by a function in two variables, g = f (x,y).

As before, to define g, we first create a set of (x,y) points over the domain of the function using the **meshgrid** command. Next, we assign the function itself. Finally, we use the **surf** command to create a surface plot.

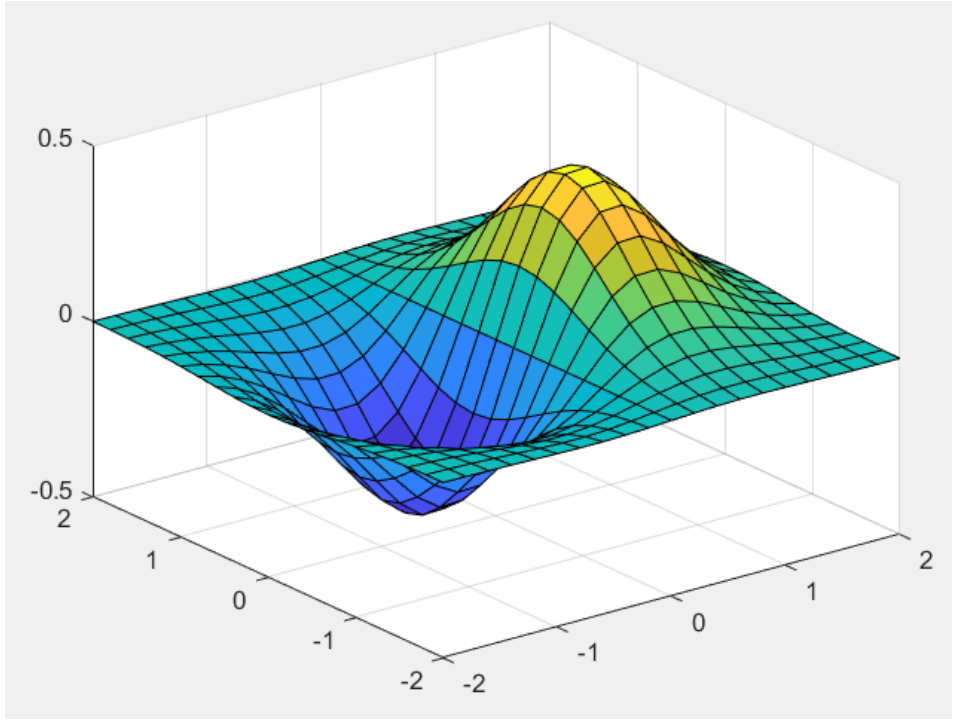
Cont.

Example:

Let us create a 3D surface map for the function $g = xe^{(-x^2-y^2)}$

```
[x,y] = meshgrid(-2:.2:2);
g = x .* exp(-x.^2 - y.^2);
surf(x, y, g)

title('Testing')
xlabel('X')
ylabel('Y')
zlabel('g')
```

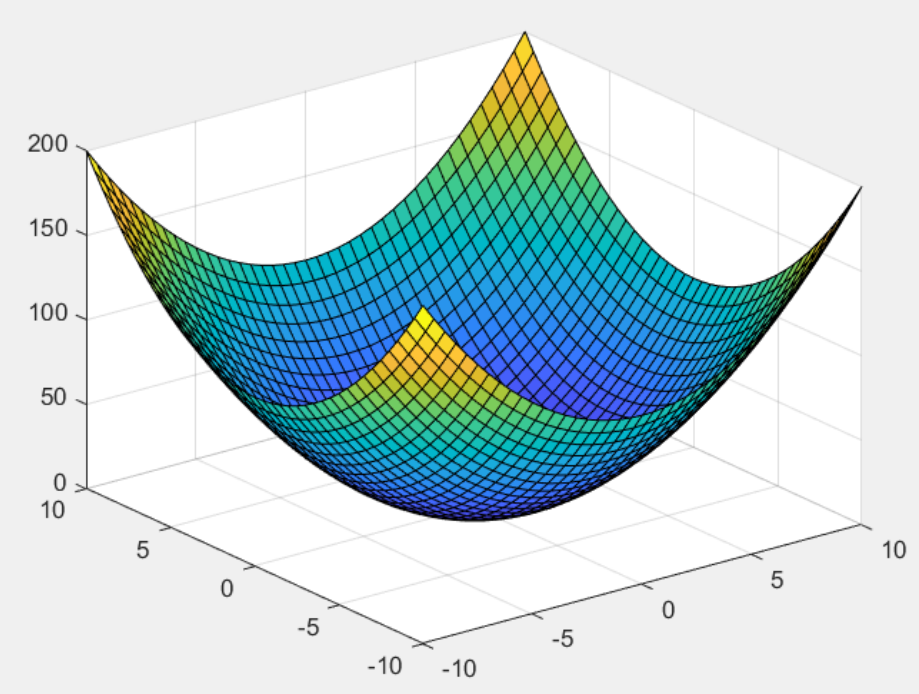


Cont.

Example:

Let us create a 3D surface map for the function f= $x^2 + y^2$

```
[x,y] = meshgrid(-10:.5:10);
f=x.^2 + y.^2;
surf(x, y, f)
```



Questions/queries?