SEMESTER – I

19PHU113 **MATHEMATICS PRACTICAL – I** 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Internal: 40 External: 60 Total: 100

End Semester Exam: 3 Hours

Course Objectives

This course enables the students

- To develop skills for quantitative estimation using computer language.
- To code various differentiation and integration methods in a modern computer language.
- To plot the graphs of function

Course Outcomes (COs)

On successful completion of this course, the student will be able to

- Solve complicated matrix related problems like matrix inverse and matrix multiplication.
- Acquire problem-solving skills through computer programming.
  Plot various functions and parametric curves.

List of Practical

1. Matrix addition.
2. Matrix multiplication.
3. Inverse of a matrix.
4. Transpose of a matrix
5. Plotting of graphs of function e^{ax+b}, log(ax + b), 1/(ax + b), sin(ax + b), cos(ax + b), |ax +b| and to illustrate the effect of a and b on the graph.
6. Plotting the graphs of polynomial of degree 4 and 5 and the derivative graph.
7. Sketching parametric curves. (Eg. Circle, Ellipse, Cycloid and Asteroid).
8. Evaluating definite integrals.(Line integral)

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari (Po),
Coimbatore –641 021

CLASS: I B.Sc Physics **COURSENAME: Mathematics Practical-I**

COURSE CODE: 19PHU113 **BATCH-2019-2022**

LAB MANUAL

CONTENTS

EX. NO : 1

MATRIX ADDITION

Question:

Add any two matrices of order three using Scilab.

Aim:

To add the two matrices of order three using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2: Define the matrices A and B of order three.

Step 3: Add the matrices A and B and store the value in matrix C.

Step 4: Save and execute the programme.

Step 5: Run the programme and view the output in Scilab console.

Step 6: Stop the programme.

Coding:

```
// Matrix Addition

clc

A = [ 1 2 3; 4 5 6 ; 7 8 9 ];

B= [0 4 5; 7 6 8 ; 1 2 3];

C = A+B
```

Output:

```
-->C

C =

    1.    6.    8.

   11.   11.   14.

    8.   10.   12.
```

Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 2

MATRIX MULTIPLICATION

Question:

Multiply any two matrices of order three using Scilab.

Aim:

To multiply the two matrices of order three using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2: Define the matrices A and B of order three.

Step 3: Multiply the matrices A and B and store the value in matrix C.

Step 4: Save and execute the programme.

Step 5: Run the programme and view the output in Scilab console.

Step 6: Stop the programme.

Coding:

```
// Matrix Multiplication

clc

A = [ 1 2 3; 4 5 6 ; 7 8 9 ];

B= [0 4 5; 7 6 8 ; 1 2 3];

C = A *B
```

Output:

```
-->C

C =

    17.    22.    30.

    41.    58.    78.

    65.    94.    126.
```

Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 3

TRANSPOSE OF A MATRIX

Question:

Transpose of any matrix of order three using Scilab.

Aim:

To transpose the matrix of order three using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2: Define the matrix A of order three.

Step 3: Transpose the matrix A and store the value in matrix B.

Step 4: Save and execute the programme.

Step 5: Run the programme and view the output in Scilab console.

Step 6: Stop the programme.

Coding:

```
// Transpose of a matrix
clc
A = [ 1 2 3; 4 5 6 ; 7 8 9 ];
B = A'
```

Output:

```
-->B

B =

    1.    4.    7.

    2.    5.    8.

    3.    6.    9.
```

Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 4

INVERSE OF A MATRIX

Question:

Find the inverse of a matrix of order three using Scilab.

Aim:

To find the inverse of a matrix of order three using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2: Define the matrix A of order three.

Step 3: Inverse the matrix A and store the value in matrix B.

Step 4: Save and execute the programme.

Step 5: Run the programme and view the output in Scilab console.

Step 6: Stop the programme.

Coding:

```
// Inverse of a matrix
clc
A = [ 1 2 3; 4 5 6 ; 7 8 9 ];
B = inv(A)
```

Output:

matrix is close to singular or badly scaled. rcond = 1.5420E-18

-->B

B =

-4.504D+15	9.007D+15	-4.504D+15.
9.007D+15	-1.601D+16	9.007D+15
-4.504D+15	9.007D+15	-4.504D+15

Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 5

Plotting of graphs of functions e^{ax+b} , $\log(ax+b)$, $\frac{1}{ax+b}$, $\sin(ax+b)$ $\cos(ax+b)$, $|ax+b|$ and to illustrate the effect of a and b on the graph

Question:

Plot exponential, logarithmic, trigonometric functions using Scilab.

Aim:

To plot exponential, logarithmic, trigonometric functions using Scilab**.**

Algorithm:

Step 1: Start the programme.

Step 2: Give the values of a and b by real numbers.

Step 3: Define the domain of the function and store it in x.

Step 4: Store the plotting function in y.

Step 5: Plot the function y and give the title of the graph.

Step 6: Save and execute the programme.

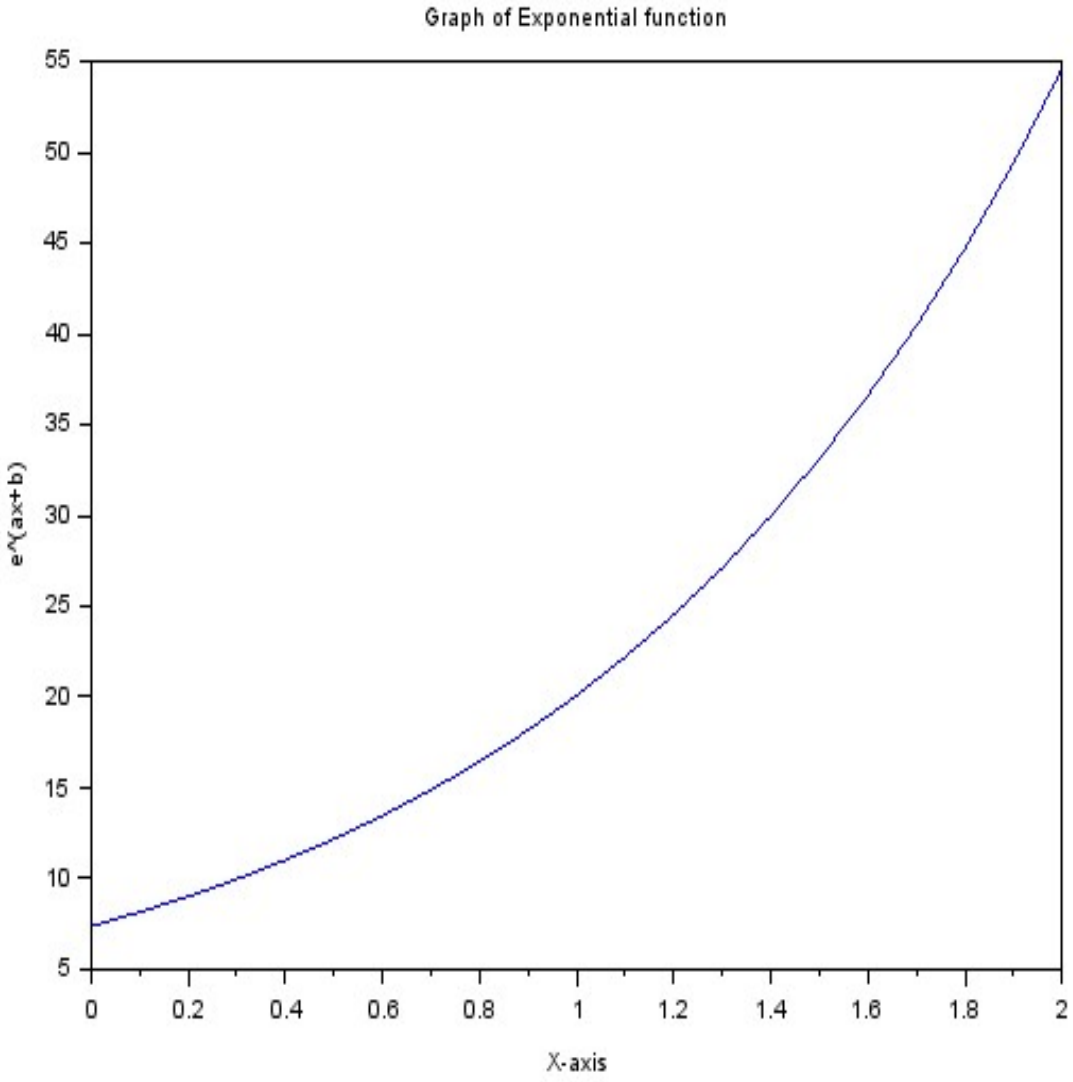
Step 7: Run the programme and view the output in graphic window.

Step 8: Stop the programme.

Coding:

```
// Plotting of e^(ax+b)

clf

a = 1;

b = 2;

x = [ 0 : 0.1  : 2];

y =  exp(a*x +b);

plot(x, y);

title(" Graph of Exponential Function")
xlabel ("X-axis")
ylabel (" e^(ax+b)" )
```

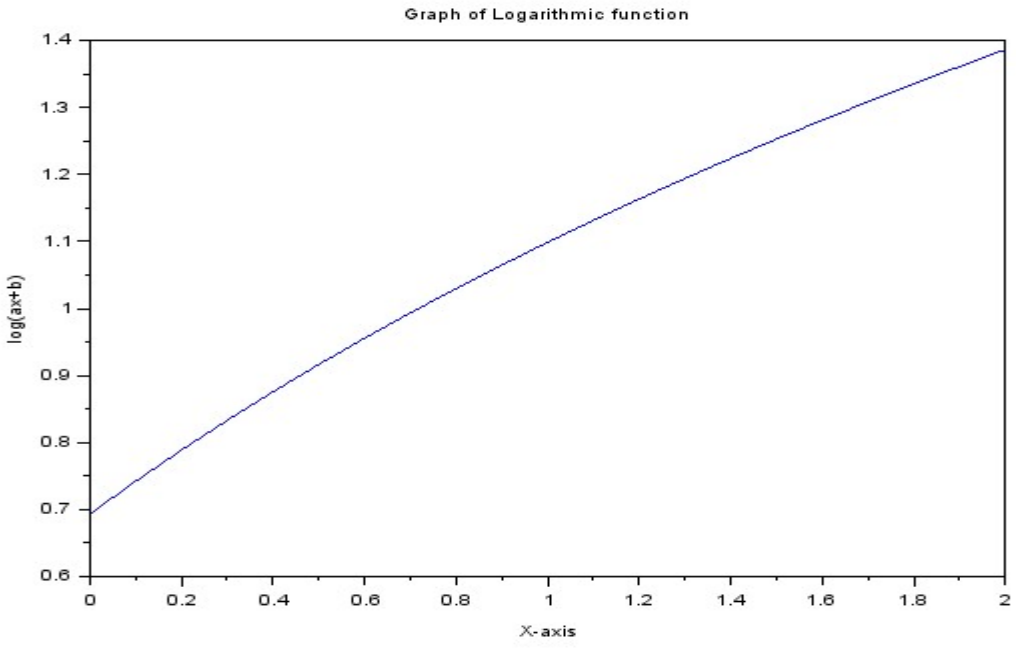
Output:


Graph of Exponential function
e^(ax+b)
X-axis
55
50
45
40
35
30
25
20
15
10
5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Coding:

```
// Plotting of log (ax + b)

clf

a = 1;

b = 2;

x = [ 0 : 0.1  : 2];

y =  log(a*x +b);

plot(x, y);

title(" Graph of Logarithmic Function")
xlabel ("X-axis")
ylabel (" log(ax+b)" )
```

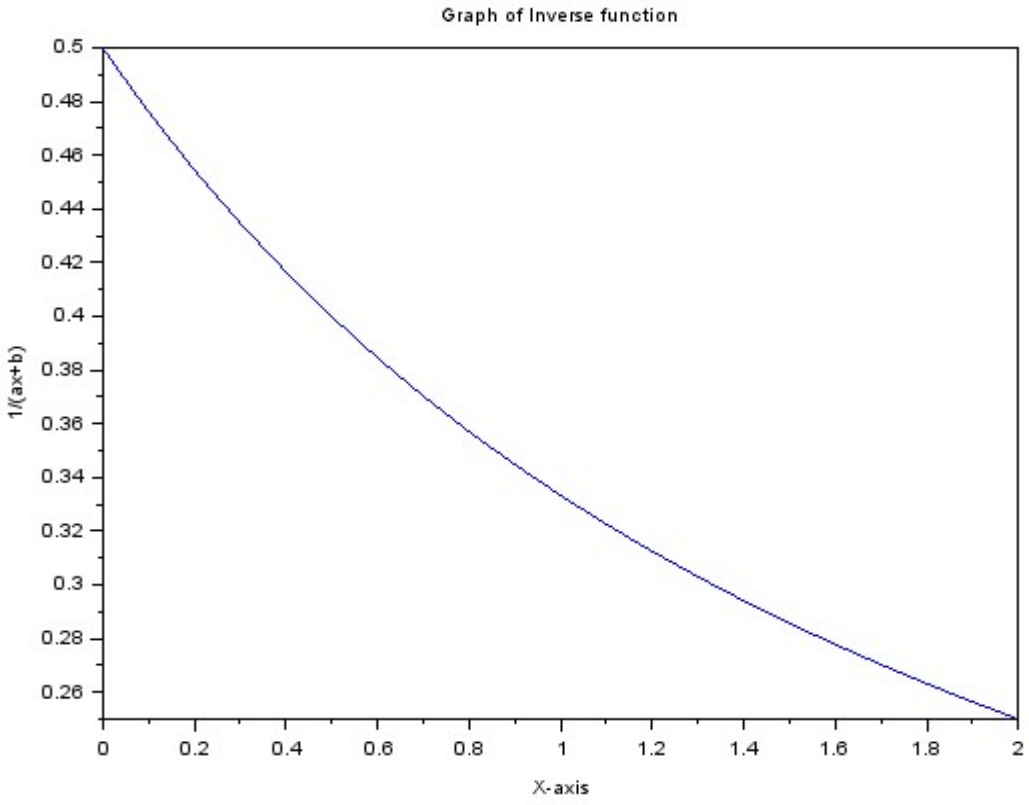
Output:



Coding:

```
// Plotting of 1/(ax+b)

clf

a = 1;

b = 2;

x = [ 0 : 0.1  : 2];

y =  (a*x+b)^-1;

plot(x, y);

title(" Graph of Inverse Function")
xlabel ("X-axis")
ylabel (" 1/ ax+b " )
```

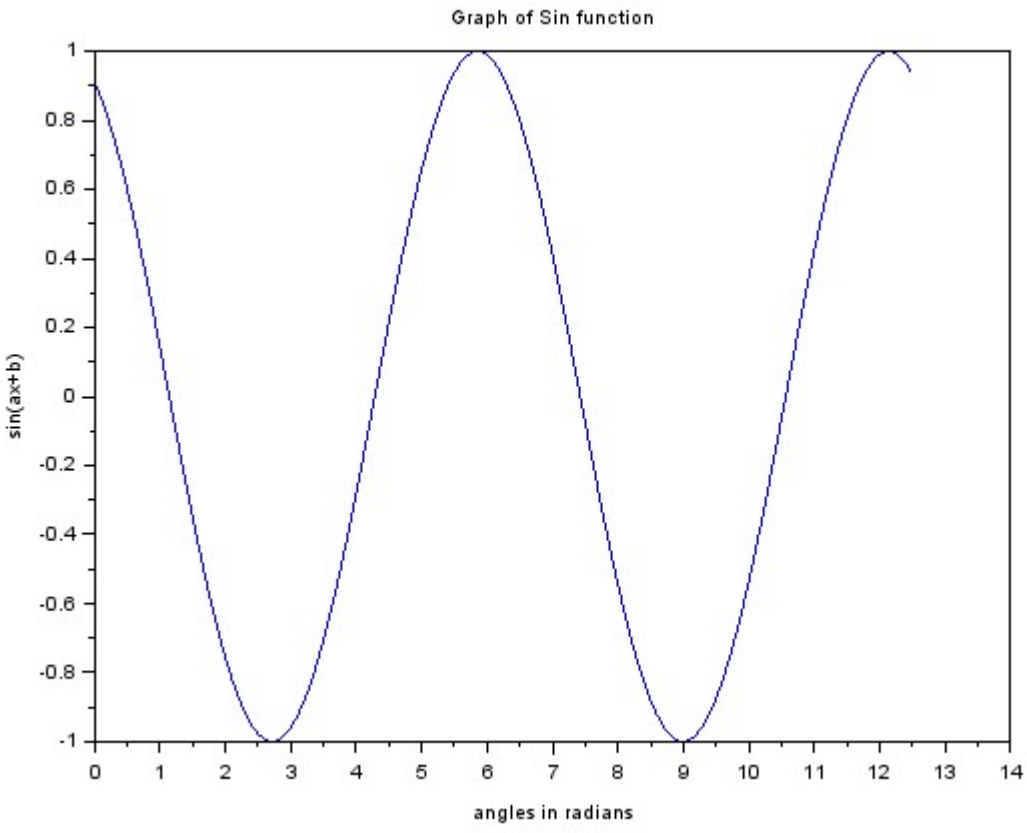
Output:



Coding:

```
// Plotting of sin (ax + b)

clf

a = 1;

b = 2;

x = [ 0 : 0.1  : 2*%pi];

y =  sin(a*x+b);

plot(x, y);

title(" Graph of Sin Function")
xlabel ("angles in radians")
ylabel (" sin(ax+b)" )
```

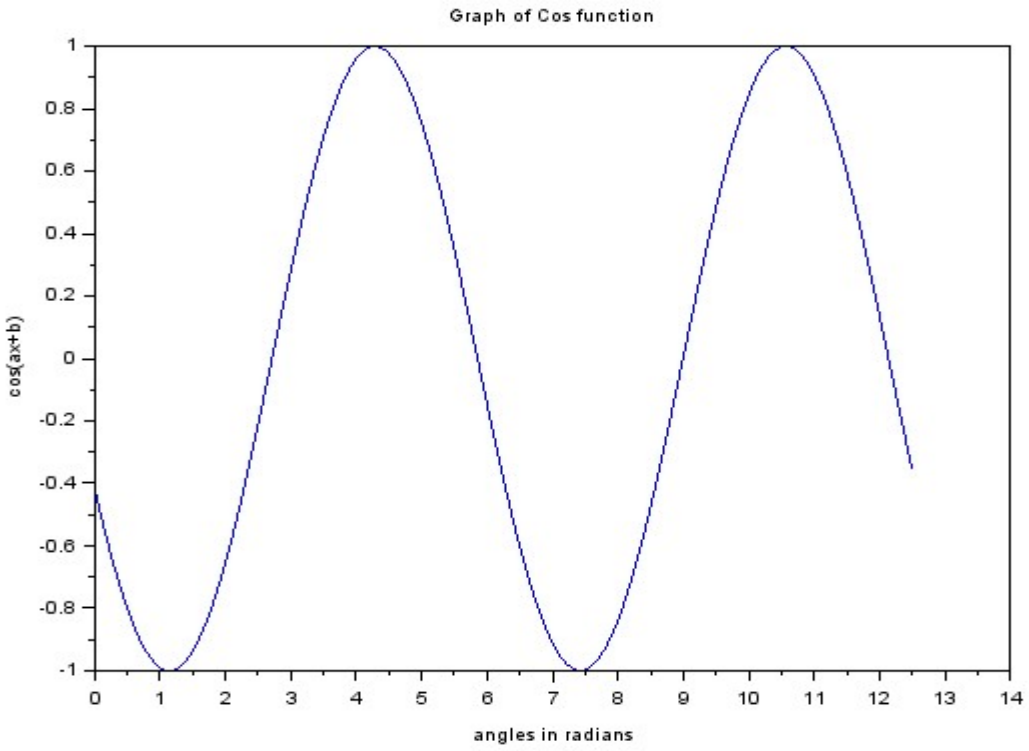
Output:



Coding:

```
// Plotting of cos (ax + b)

clf

a = 1;

b = 2;

x = [ 0 : 0.1  : 2*%pi];

y =  cos(a*x+b);

plot(x, y);

title(" Graph of Cos Function")
xlabel ("angles in radians")
ylabel (" cos(ax+b)" )
```

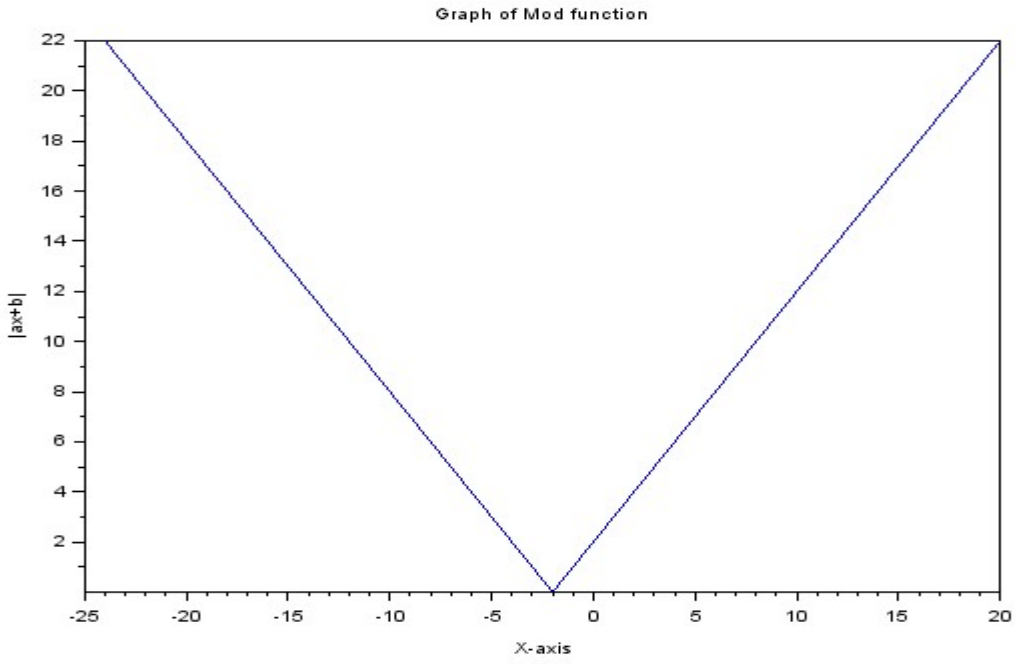
Output:



Coding:

```
// Plotting of |ax + b|

clf

a = 1;

b = 2;

x = [ -24 : 0.1  : 20];

y =  abs(a*x+b);

plot(x, y);

title(" Graph of Mod Function")
xlabel ("X-axis")
ylabel (" |ax+b|" )
```

Output:



Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 6

Plotting of graphs of polynomials of degree 4 and 5 and the derivative graph

Question:

Plot the graphs of polynomials of degree 4 and 5 and also its derivative using Scilab.

Aim:

To plot the graphs of polynomials of degree 4 and 5 and also its derivative using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2: Use linspace command to fix the values of x.

Step 3: Define the polynomials p1(x) and p2(x).

Step 4: Display the values of p1(x) and p2(x)

Step 4: Find derivatives of p1(x) and p2(x) and store it in d1 and d2.

Step 5: Use plot2d command to plot the graphs of polynomials and its derivatives

Step 6: Save and execute the programme.

Step 7: Run the programme and view the output in graphic window.

Step 8: Stop the programme.

Coding:

```
clf

x=linspace(0, 1, 100)';

p1= 4*x^4 + 5*x^3 +2*x^2 + 6* x +7;

p2= 7*x^5 -3*x^4 +8*x^3 +2 *x^2 +4*x +9;

disp(p1);

disp(p2);

d1=derivat(p1);

d2=derivat(p2);
```

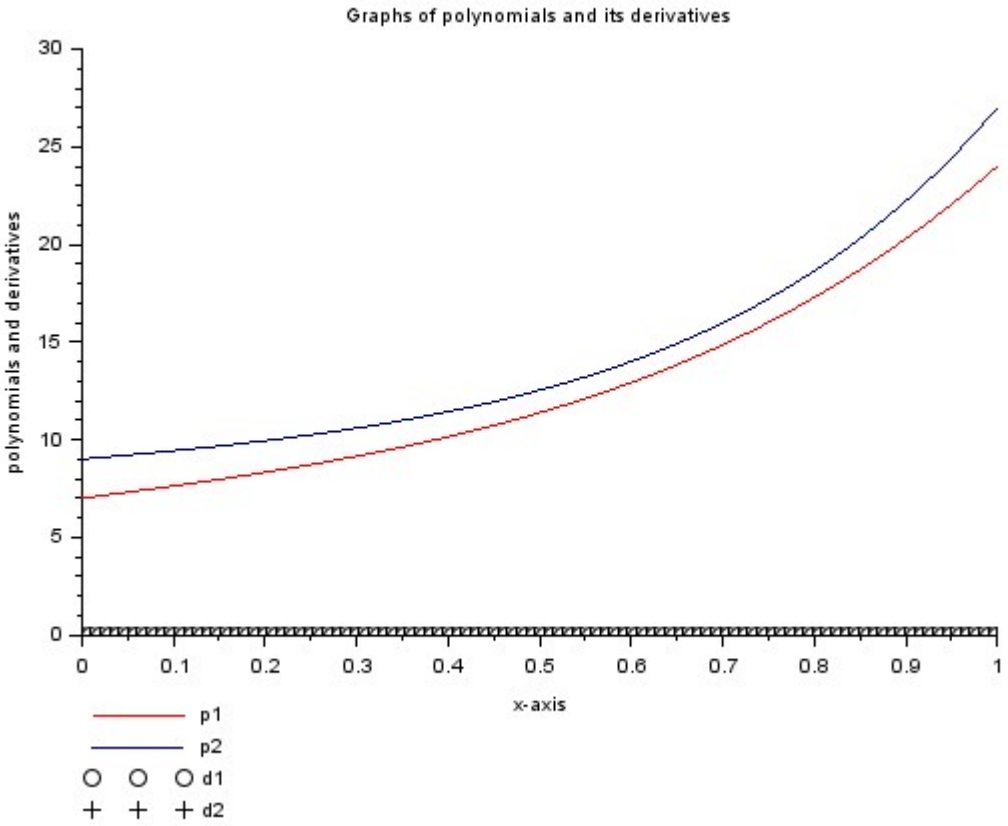
plot2d(x, [p1 p2 d1 d2], [5 9 -9 -1], leg=”p1@p2@d1@d2”);

title(“ Graphs of polynomials and its derivatives “)

xlabel(“ X-axis”)

ylabel(“ Polynomials and derivatives”)

Output:



Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 7

Sketching Parametric Curves (Eg. Circle, Ellipse, Cycloid and Asteroid)

Question:

Sketch the parametric curves (Eg: Circle, Ellipse, Cyclod and Asteroid)

Aim:

To sketch the parametric curves (Eg: Circle, Ellipse, Cycloid and Asteorid)

Algorithm:

Step 1: Start the programme.

Step 2 : Use linspace command to fix the values of t.

Step 3: Define the parametric curves.

Step 4: Use plot command to plot the parametric curves.

Step 5: Save and execute the programme.

Step 6: Run the programme and view the output in graphic window.

Step 7: Stop the programme.

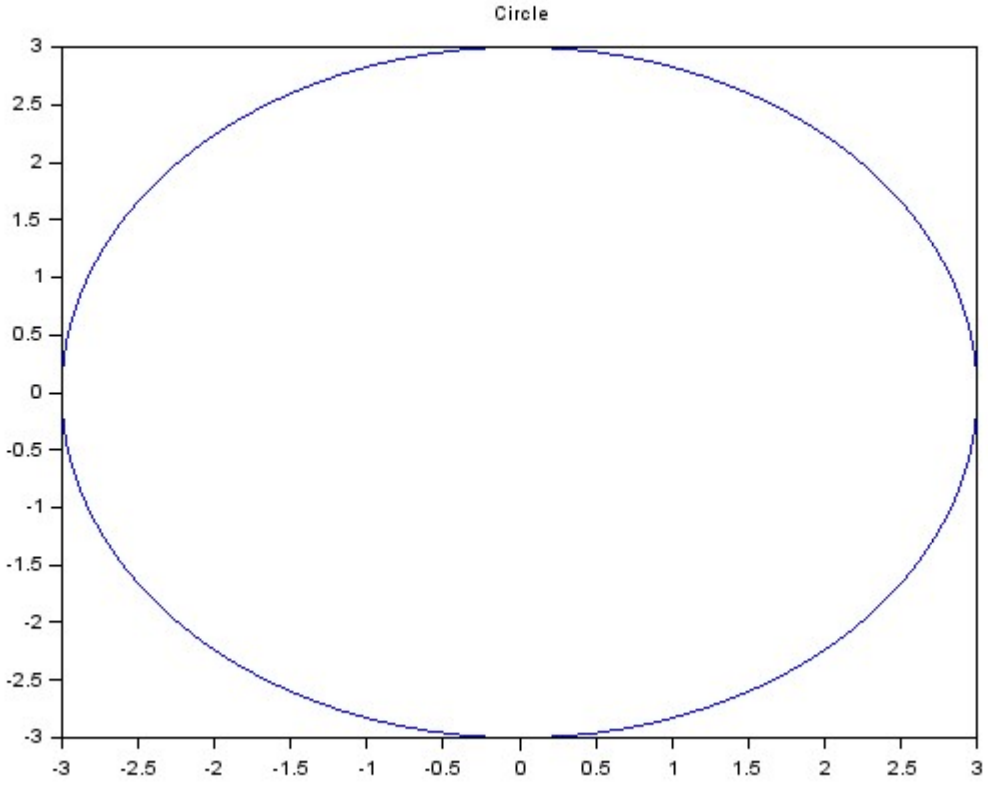
Coding:

a) **Circle**

```
clf;

r = 3;

t = linspace(0, 2*%pi, 100);
```

```
x = r * cos(t);

y = r*sin(t);

plot(x,y);

title("Circle")
```

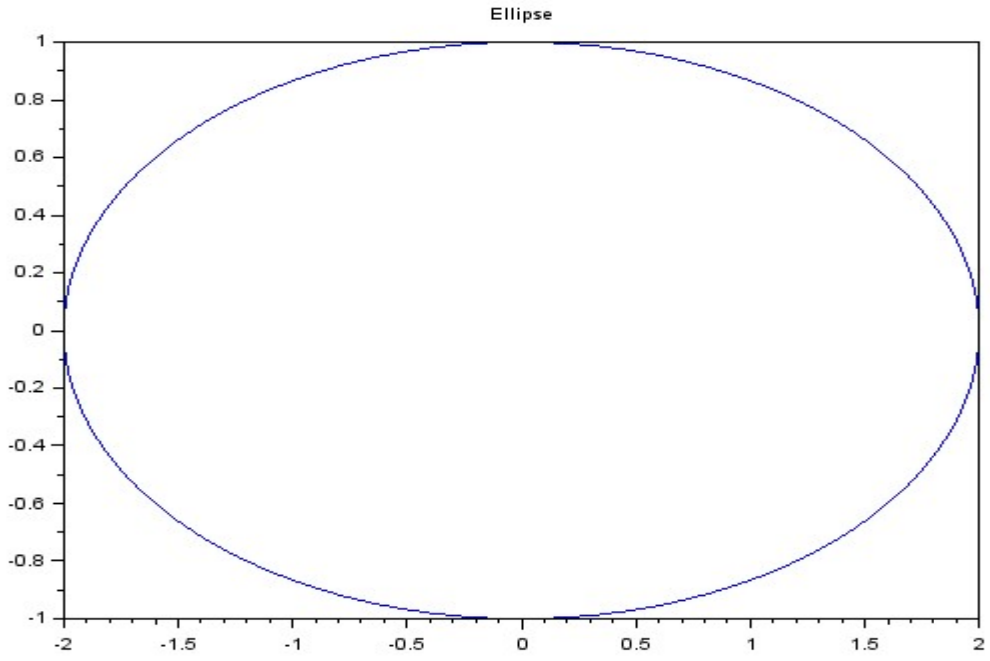
Output:



b) Ellipse

```
clf

a = 2;

b = 1;

t = linspace(0, 2*%pi,100);

x = a*cos(t);

y = b*sin(t);

plot(x,y);

title("Ellipse")
```

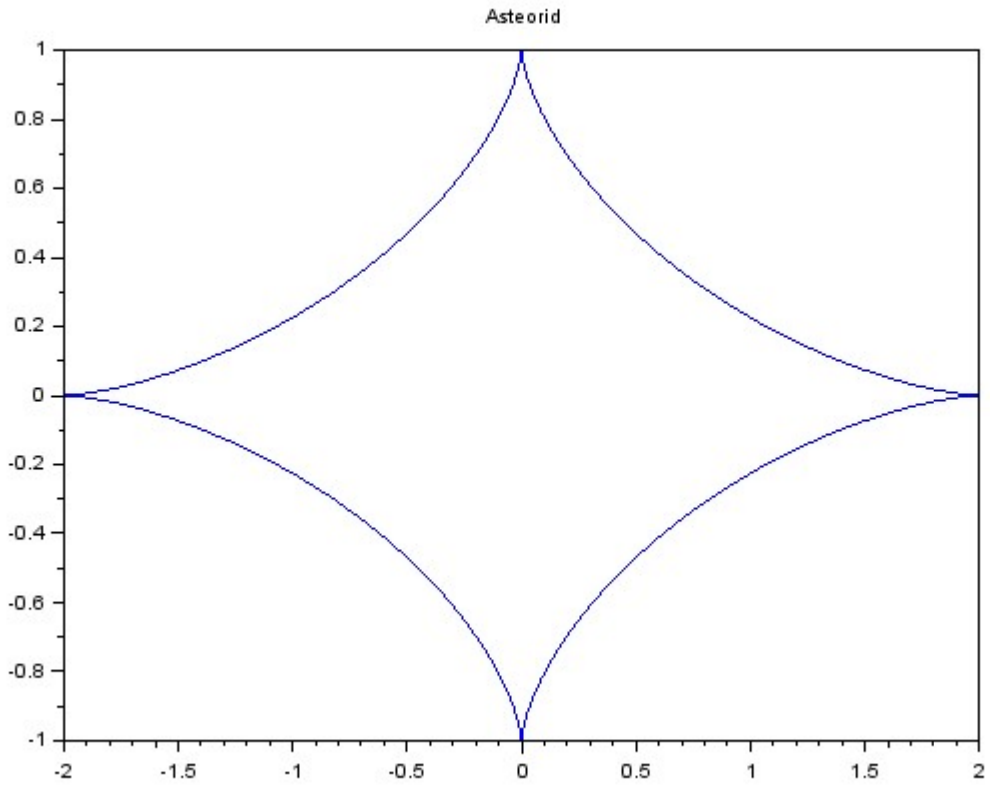
Output:



c) Asteroid

```
clf

a = 2;

t = linspace(0, 2*%pi, 100);

x = a*cos(t)^3;

y = a*sin(t)^3;

plot(x,y);

title("Asteroid")
```

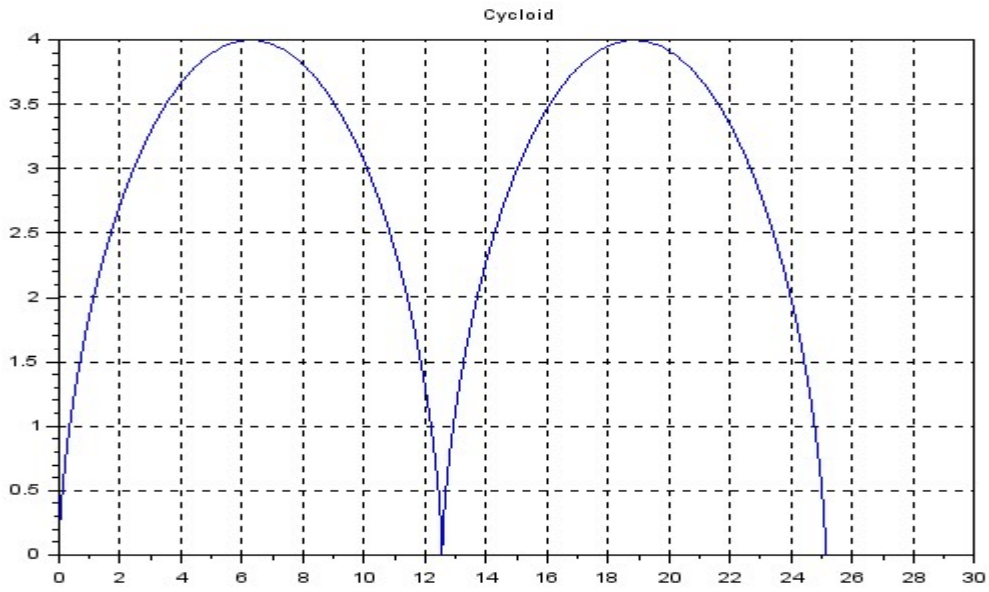
Output:



d) Cycloid

```
clf

a = 2;

t = linspace(0, 4*%pi, 100);

x = a*(t-sin(t));

y = a*(1-cos(t));

plot(x,y);

title("Cycloid")
```

Output:



Result:

Thus the programme has been executed successfully and the output has been verified.

EX. NO : 8

Evaluating definite integral (Line integral)

Question:

Evaluate the given definite integral using Scilab.

Aim:

To evaluate the definite integral using Scilab.

Algorithm:

Step 1: Start the programme.

Step 2 : Enter the lower limit and upper limit in a and b.

Step 3: Use integrate command to evaluate the given definite integral.

Step 4: Display the integral value

Step 5: Save and execute the programme.

Step 6: Run the programme and view the output in console.

Step 7: Stop the programme.

Coding:

```
clc
a  =  input("Enter the lower limit");
b  = input("Enter the upper limit");
I = integrate('6*x^2 -7*x +2', 'x', a, b);
disp(I)
```

Output:

Enter the lower limit0

Enter the upper limit2

6.

--->