Roll No. 232

USN.

S.D.M.E. Society's

S.D.M. COLLEGE OF ENGINEERING & TECHNOLOGY

DHARWAD - 580 002

Department of ECE

LABORATORY CERTIFICATE

This is to Certify that Sri/Kum AKSHATA . M . GOUDAR

has satisfactorily completed the course of experiments in

practical prescribed by the S.D.M. College of Engineering & Technology, Dharwad for .... *Semester in the Laboratory of this College during the year 20 - 20*

Place :

Date :

Staff in-charge **H.O.D.** **Principal**

S.D.M.E. Society's

S.D.M. COLLEGE OF ENGINEERING & TECHNOLOGY

DHARWAD - 580 002

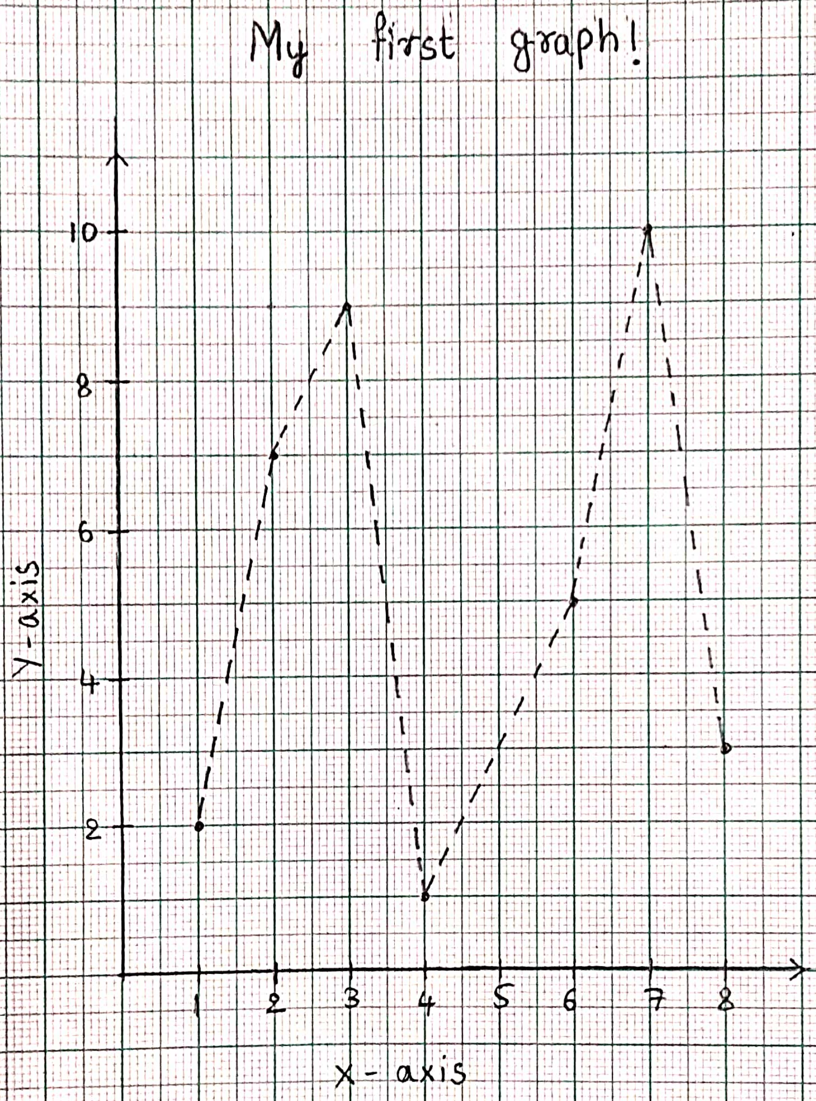
CONTENTS

Head of the Department

Staff Member in-charge

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :



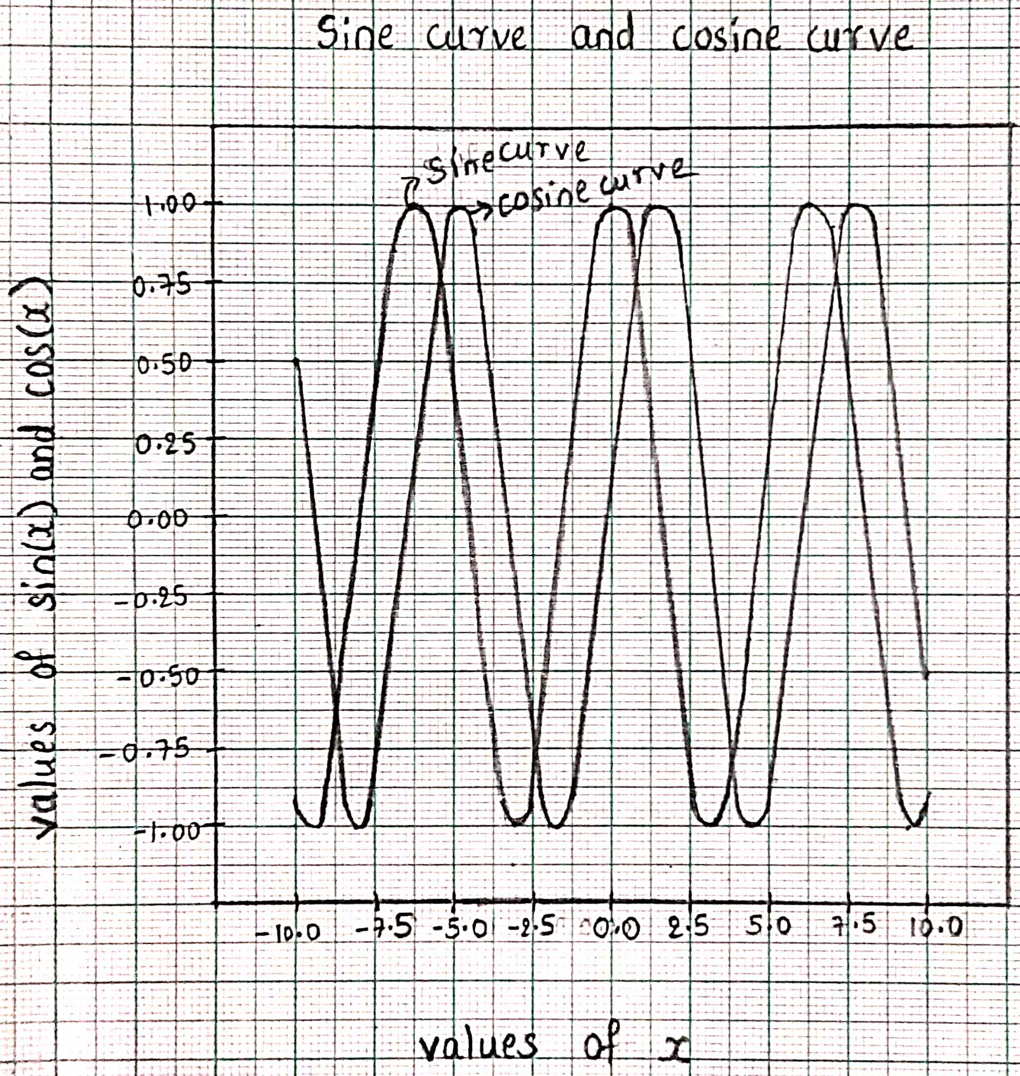
2D Plots of Cartesian and Polar Curves

1) A program to plot a line joining the points (1,2), (2,7), (3,9), (4,1), (6,5), (7,10), (8,3).

```
import matplotlib.pyplot as plt.
x = [1,2,3,4,6,7,8]
y = [2,7,9,1,5,10,3]
plt.plot (x, y, 'r+--')
plt.xlabel ('x-axis')
plt.ylabel ('y-axis')
plt.title ('My first graph!')
plt.show ().
```

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :



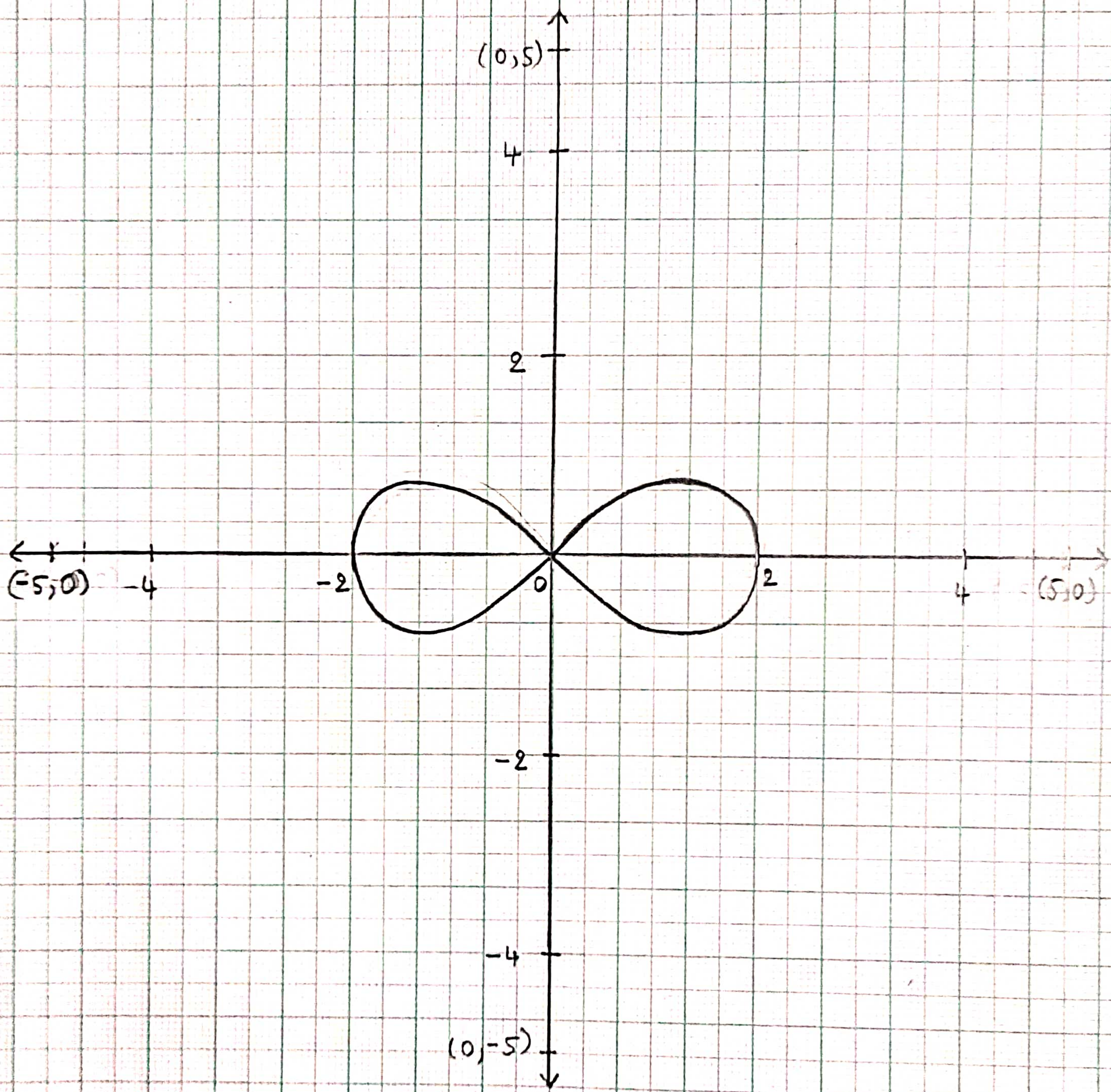
2) A program to plot a sine and cosine curves in (-10,10) with step size of 0.001.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange (-10, 10, 0.001)
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot ( x, y1 , x ,y2)
plt.title (" sine curve and cosine curve")
plt.xlabel ("values of x ")
plt.ylabel (" values of sin(x) and cos(x) ")
plt.grid ()
plt.show ()
```

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :

Lemniscate : $4y^2 = x^2(4-x^2)$



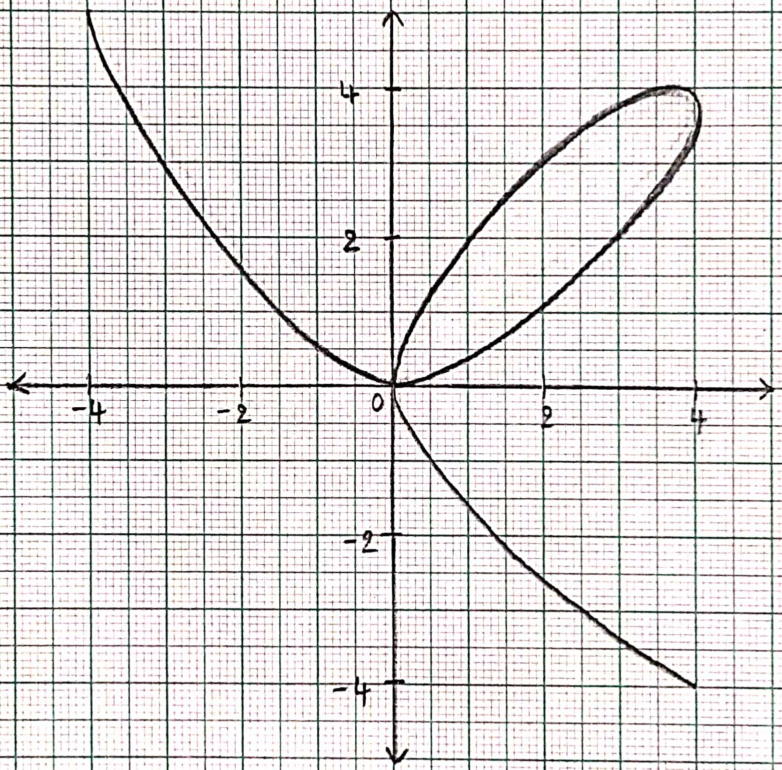
3) A program to plot an implicit curve $a^2y^2 = x^2(a^2 - x^2)$ [Lemniscate], taking $-5 \le x \le 5$; $-5 \le y \le 5$; $a = 2$.

```
from sympy import plot_implicit, Symbols, Eq
x, y = Symbols ('x ,y')
p5 = plot_implicit ( Eq ( 4*(y**2), (x**2)*(4-x**2)),
(x, -5 ,5) , (y, -5,5))
```

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :

Folium of De-Cartes : $x^3 + y^3 = 3axy$



4) A program to plot an implicit curve $x^3 + y^3 = 3axy$ [Folium of De-Cartes], taking $-5 \le x \le 5$; $-5 \le y \le 5$; $a = 2$.

```
from sympy import plot_implicit, symbols, Eq
x,y = symbols('x,y')
p6 = plot_implicit(Eq(x**3+y**3, 3*2*x*y),
(x,-5,5), (y,-5,5))
```

EXPERIMENT NO.:

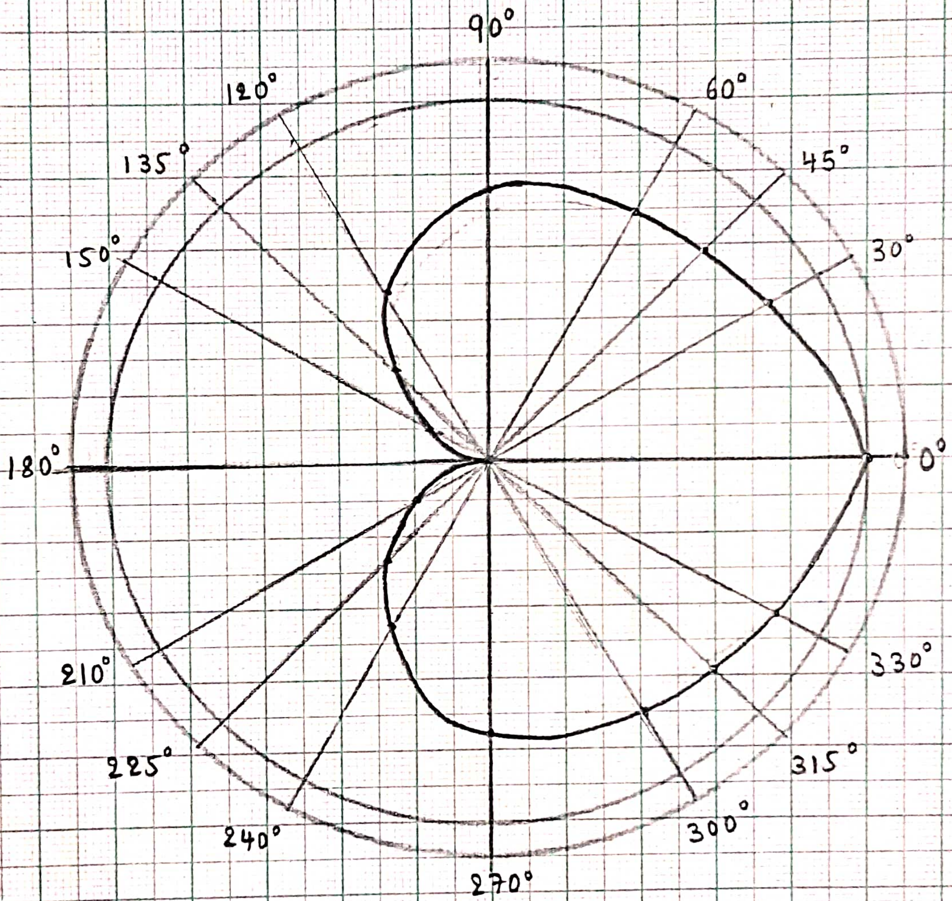
SCALE

X - axis :

DATE :

Y - axis :

Cardiod : $r = 5(1+\cos\theta)$



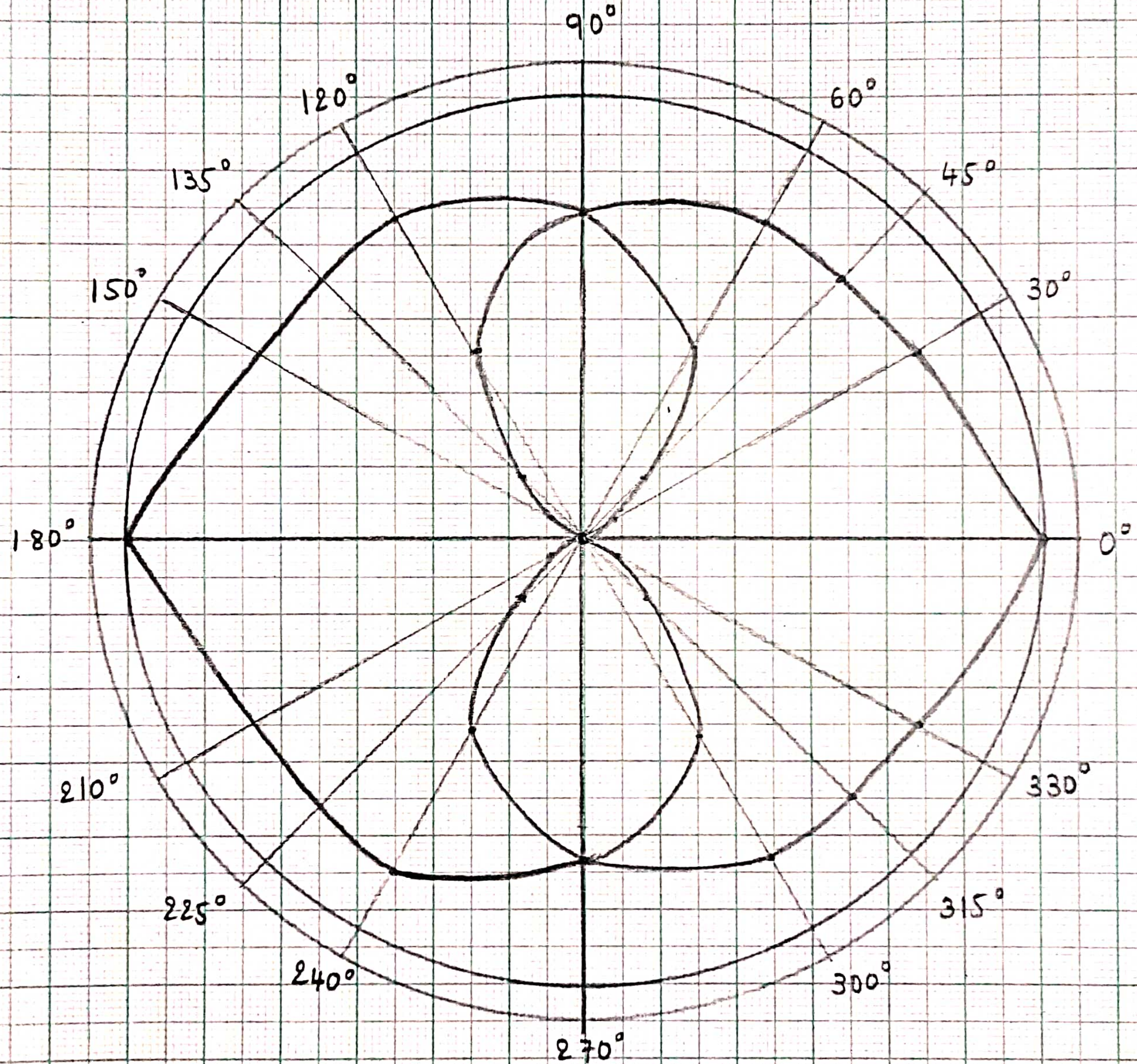
5) A program to plot a polar curve $r = 5(1+\cos\theta)$ [Cardiod], taking $0 \le \theta \le 2\pi$ with 1000 linespace

```
from pylab import *
theta = linspace (0, 2*np.pi, 1000)
r1 = 5+5 *cos(theta)
polar (theta, r1 ,'r')
show ()
```

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :

Cardiods: $r_1 = 3 + 3\cos\theta$ and $r_2 = 3 - 3\cos\theta$



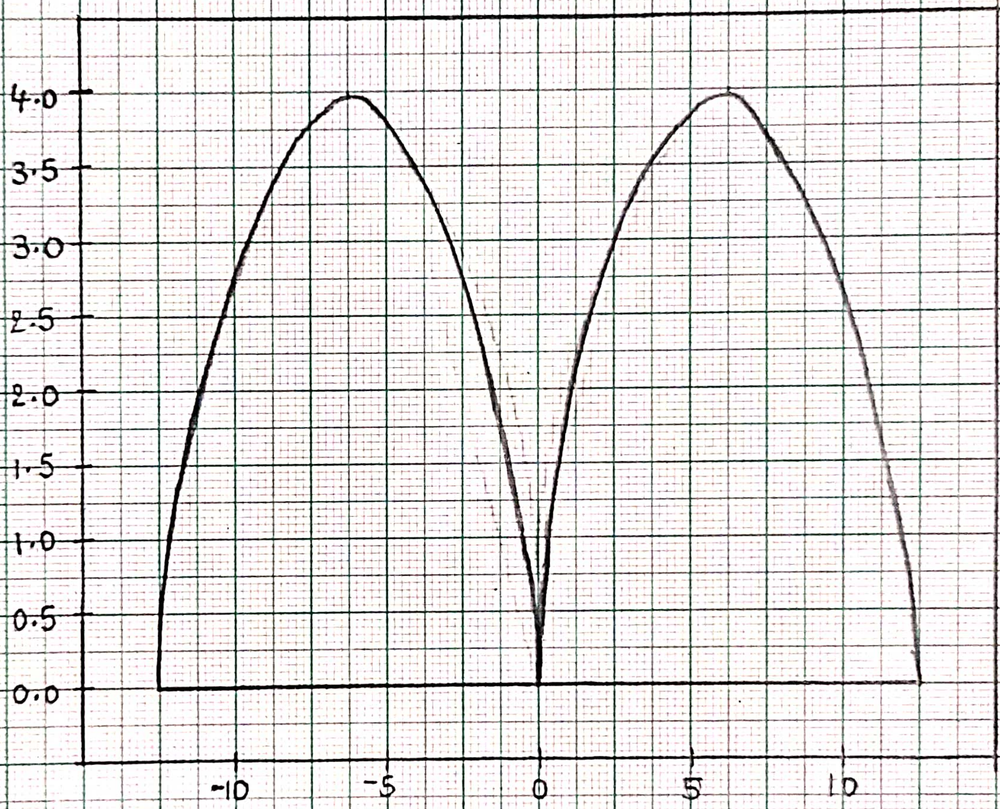
6) A program to plot the cardiods $r = a(1+\cos\theta)$ and $r = a(1-\cos\theta)$, taking $0 \le \theta \le 2\pi$ with step size of 0.01.

```
import numpy as np
import matplotlib.pyplot as plt
import math
plt.axes (projection = 'polar')
a = 3
rad = np.arange (0, (2*np.pi), 0.01)
for i in rad :
        r = a + (a* np.cos(i))
        plt.polar (i, r,'g.')
        r1 = a - (a* np. cos (i))
        plt.polar (i , r1, 'r.')
plt. show ()
```

EXPERIMENT NO.: SCALE X - axis :

DATE : Y - axis :

Cycloid: $x = a(\theta - \sin\theta)$ & $y = a(1 - \cos\theta)$



7) A program to plot the parametric curve, Cycloid $x = a(\theta - \sin\theta)$ and $y = a(1-\cos\theta)$, taking $-2\pi \leq \theta \leq 2\pi$ with 100 linespace

```
import numpy as np
import matplotlib.pyplot as plt
def cycloid (r):
    x = []
    y = []
for theta in np.linspace (-2*np.pi, 2*np.pi, 100):
    x.append (r* (theta - np.sin(theta)))
    y.append (r* (1 - np.cos(theta)))
plt.plot (x,y)
plot.show ()
cycloid (2)
```

LAB 2 : FINDING ANGLE BETWEEN TWO POLAR CURVES, CURVATURE AND RADIUS OF CURVATURE.

1) Program to find the angle between the curves $r=4(1+\cos t)$ and $r=5(1-\cos t)$.

```
from sympy import *
r,t = symbols('r,t')
r1 = 4 * (1+ cos(t))
r2 = 5 * (1-cos(t))
dr1 = diff (r1 , t)
dr2 = diff (r2 , t)
t1 = r1 / dr1
t2 = r2 / dr2
q = solve (r1 - r2 , t)
w1 = t1.subs ({t: float (q[1])})
w2 = t2.subs ({t: float (q[1])})
y1 = a tan (w1)
y2 = a tan (w2)
w = abs (y1 - y2)
print ('Angle between curves in radians is % 0.3f' % (w))
```

Output :

The radius of curvature is 2.37 units.

2) Program to find the radius of curvature of $r = 4(1+\cos t)$ at $t = \pi/2$

```
from sympy import *
t = Symbol ('t')
r = Symbol ('r')
r = 4 * (1+ cos(t))
r1 = Derivative (r,t).doit()
r2 = Derivative (r1,t).doit()
rho = (r**2 + r1**2) ** (1.5) / (r**2 + 2*r1**2 -r*r2)
rho1 = rho.subs (t, pi/2)
print ('The radius of curvature is %3.4f units' % rho1)
```

Output :

The radius of curvature is $\frac{(a^2)^{1.5}}{2a^2}$.

3) Program to find the radius of curvature of r=asin(nt) at t=pi/2 and n=1.

```
from sympy import *
t,r,a,n = Symbols('t r a n')
r = a* sin (n*t)
r1 = Derivative (r,t). doit()
r2 = Derivative (r1,t). doit()
rho = (r**2 + r1**2)**1.5 / (r**2 + r1**2 - r*r2)
rho1 = rho.subs (t, pi/2)
rho1 = rho1.subs (n,1)
print("The radius of curvature is")
display (simplify (rho1))
```

Output :

Mixed partial derivatives are equal.

LAB3: FINDING PARTIAL DERIVATIVES AND JACOBIANS OF FUNCTION OF SEVERAL VARIABLES.

1) Program to prove that the mixed partial derivatives, $u_{xy} = u_{yx}$ for $u = \exp(x)\,(x\cos(y) - y\sin(y))$

```
from sympy import *
x,y = symbols ('x y')
u = exp(x) * (x* cos(y) - y* sin(y))
dux  = diff (u,x)
duy  = diff (u,y)
duxy = diff (dux,y)
duyx = diff (duy,x)
if duxy== duyx :
    print ('Mixed partial derivatives are equal')
else :
    print ('Mixed partial derivatives are not equal')
```

Output :

$(x\cos(y) - y\sin(y))e^{x}$

Ans : 0.0

2) Program to prove that if $u = exp(x)(x\cos(y) - y\sin(y))$, then $u_{xx} + u_{yy} = 0$.

```
from sympy import *
x,y = symbols ('x y')
u = exp(x) * (x * cos(y) - y * sin(y))
display (u)
dux = diff (u,x)
duy = diff (u,y)
uxx = diff (dux,x)
uyy = diff (duy,y)
w = uxx + uyy
w1 = simplify(w)
print ('Ans :' float (w1)).
```

Output :

The Jacobian matrix is

$$\begin{bmatrix} 1 & 6y & 3z^2 \\ 8xyz & 4x^2z & 4x^2y \\ -y & -x & 6z^2 \end{bmatrix}$$

$$J = 4x^3y - 24x^2y^3 + 12x^2z^3y + 24x^2z^3 - 288xy^2z^3$$

J at $(1,-1,0)$: 20

3) Program to prove that at $(1,-1,0)$, $J = 20$ if $u = x+3y^2-z^3$, $v = 4x^2yz$, $w = 2z^2 - xy$.

```
from sympy import *
x,y,z = symbols('x,y,z')
u = x + 3*y**2 - z**3
y = 4*x**2*y*z
w = 2*z**2 - x*y.
dux = diff (u,x)
duy = diff (u,y)
duz = diff (u,z)
dvx = diff (v,x)
dvy = diff (v,y)
dvz = diff (v,z)
dwx = diff (w,x)
dwy = diff (w,y)
dwz = diff (w,z)
J = Matrix ([[dux ,duy,duz], [dvx ,dvy,dvz], [dwx,dwy,dwz]])
print ("The Jacobian matrix is")
display (J)
Jac = Determinant (J).doit()
print ('\n\n J = \n')
display (Jac)
J1 = J.subs ([(x,1) ,(y,-1) ,(z,0)])
print ('\n\n J at (1,-1,0):\n')
Jac1 = Determinant (J1) .doit ()
display (Jac 1)
```