**Sahakar Maharshi Bhausaheb Santuji Thorat College Sangamner**

**Remark Demonstrator’s Signature**

**Date:- / /20**

**DEPARTMENT OF COMPUTER SCIENCE**

**Sub : Mathematics**

**Name:-\_Gorde Yash Somnath Roll.No:-\_21 Date:- Title of the expt:- Slip no 19 Page.no:- Class:- BCS**

# Q1 . Attempt any TWO of the following

**A ) Write a python program to plot the graph of sin(x),ex and x3 in [0,5] in one figure with 2x2 subplots**

**-**

import numpy as np

import matplotlib.pyplot as plt x = np.linspace(0, 5, 100)

y1 = np.sin(x) y2 = np.exp(x) y3 = x\*\*3

fig, axs = plt.subplots(2, 2)

axs[0, 0].plot(x, y1) axs[0, 0].set\_title('sin(x)') axs[0, 1].plot(x, y2)

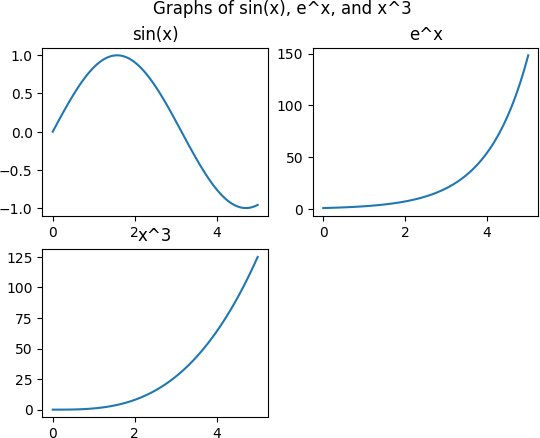
axs[0, 1].set\_title('e^x')

axs[1, 0].plot(x, y3)

axs[1, 0].set\_title('x^3')

axs[1, 1].axis('off')

fig.suptitle('Graphs of sin(x), e^x, and x^3') plt.show()



# B ) write python program to plot 3D surface plot of the function z=cos([x]+[y]) in - 1<x,y<1

**-**

import numpy as np

import matplotlib.pyplot as plt

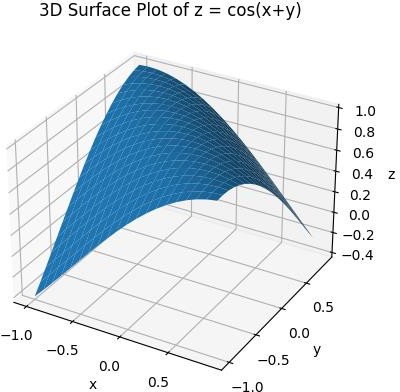
from mpl\_toolkits.mplot3d import Axes3D x = np.arange(-1, 1, 0.1)

y = np.arange(-1, 1, 0.1) X, Y = np.meshgrid(x, y) Z = np.cos(X + Y)

fig = plt.figure()

ax = fig.gca(projection='3d') surf = ax.plot\_surface(X, Y, Z) ax.set\_xlabel('x') ax.set\_ylabel('y') ax.set\_zlabel('z')

ax.set\_title('3D Surface Plot of z = cos(x+y)') plt.show()



# C ) Write a python program to plot 2D graph of the function f(x)=log(x)+5 and g(x)=log(x)-5 in [0,10] by setting different line width and different colors ot the curve

**-**

import numpy as np

import matplotlib.pyplot as plt x = np.linspace(0.1, 10, 100)

f = np.log(x) + 5

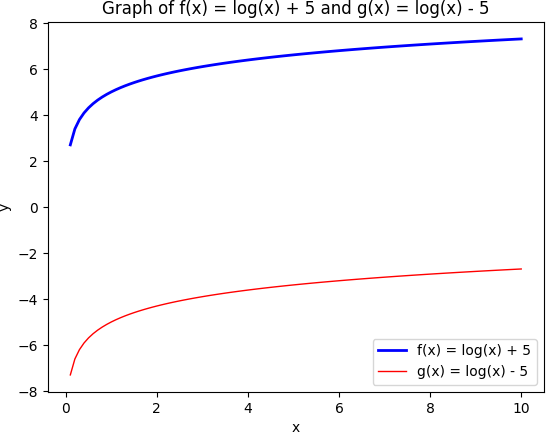
g = np.log(x) - 5

plt.plot(x, f, linewidth=2, color='blue', label='f(x) = log(x) + 5') plt.plot(x, g, linewidth=1, color='red', label='g(x) = log(x) - 5')

plt.legend() plt.xlabel('x')

plt.ylabel('y')

plt.title('Graph of f(x) = log(x) + 5 and g(x) = log(x) - 5') plt.show()



# Q2 ) Attempt any TWO of the following

**A ) Write a python program to rotate the ray by 90⸰ in clockwise direction having starting point(0,0) and end point (4,4)**

**-**

import numpy as np start = np.array([0, 0]) end = np.array([4, 4])

theta = np.pi/2 # 90 degrees in radians

R = np.array([[np.cos(theta), -np.sin(theta)],

[np.sin(theta), np.cos(theta)]]) new\_end = R.dot(end - start) + start

print(f"The new end point is ({new\_end[0]:.2f}, {new\_end[1]:.2f})")

output :

The new end point is (-4.00, 4.00)

# B ) Write a python program to reflect the triangle ABC through the line y=3 where A[1,0],B[2,-1], and C[-1,3]

**-**

import numpy as np

A = np.array([1, 0])

B = np.array([2, -1])

C = np.array([-1, 3])

L = np.array([0, 3])

R = np.array([[1, 0],

[0, -1]])

A\_new = R.dot(A - L) + L B\_new = R.dot(B - L) + L C\_new = R.dot(C - L) + L

print(f"The new vertex A is ({A\_new[0]:.2f}, {A\_new[1]:.2f})")

print(f"The new vertex B is ({B\_new[0]:.2f}, {B\_new[1]:.2f})")

print(f"The new vertex C is ({C\_new[0]:.2f}, {C\_new[1]:.2f})")

output :

The new vertex A is (1.00, 6.00) The new vertex B is (2.00, 7.00) The new vertex C is (-1.00, 3.00)

**c) Write a python program to draw a polygon with vertices (0,0),(1,0),(2,2),(1,4), Also find area and perimeter of the player**

**-**

import matplotlib.pyplot as plt vertices = [(0, 0), (1, 0), (2, 2), (1, 4)]

polygon = plt.Polygon(vertices, closed=True) fig, ax = plt.subplots() ax.add\_patch(polygon)

ax.set\_xlim(-1, 3)

ax.set\_ylim(-1, 5) ax.set\_xlabel("x") ax.set\_ylabel("y")

ax.set\_title("Polygon with Vertices (0,0), (1,0), (2,2), (1,4)")

area = 0.5 \* abs(sum(x0\*y1 - x1\*y0 for ((x0, y0), (x1, y1)) in zip(vertices, vertices[1:] + [vertices[0]])))

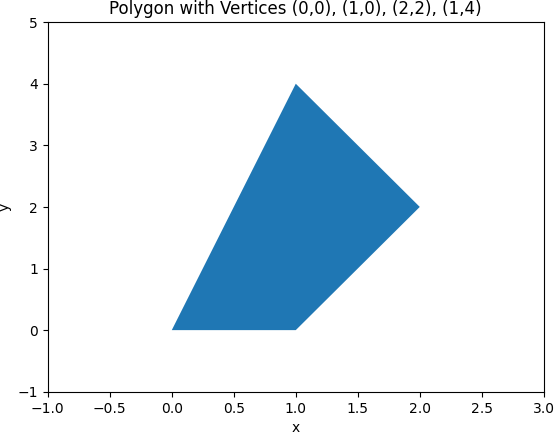
perimeter = sum(((x1-x0)\*\*2 + (y1-y0)\*\*2)\*\*0.5 for ((x0, y0), (x1, y1)) in zip(vertices, vertices[1:] + [vertices[0]]))

print(f"The area of the polygon is {area:.2f}") print(f"The perimeter of the polygon is {perimeter:.2f}") plt.show()

output :

The area of the polygon is 4.00

The perimeter of the polygon is 9.60



**Q3 ) Attempt the following**

**A ) Attempt any ONE of the following**

**I ) Write a python program to solve the following LPP : Max Z=3x+5y=4z**

**Subject to 2x+3y≤8 2y+5z≤10**

**3x+2y+4z≤15**

**X,y≥0**

-

import scipy.optimize as opt c = [-3, -5, -4]

A = [[2, 3, 0], [0, 2, 5], [3, 2, 4]]

b = [8, 10, 15]

bounds = [(0, None), (0, None), (0, None)]

res = opt.linprog(c=c, A\_ub=A, b\_ub=b, bounds=bounds)

print(f"Optimal Solution: {res.x}") print(f"Optimal Objective Value: {-res.fun}")

output :

Optimal Solution: [2.17073171 1.2195122 1.51219512]

Optimal Objective Value: 18.65853658536585

# II ) Write a python program to solve the following LPP : Min Z=x+2y+z

**Subject to x+**𝟏 𝒚 **+**𝟏 𝒛 **≤1**

𝟐 𝟐

𝟑 𝒙 **+ 2y +z≥8**

𝟐

# X,y≥0

**-**

import scipy.optimize as opt c = [1, 2, 1]

A = [[1, 1/2, 1/2], [-3/2, -2, -1]]

b = [1, -8]

bounds = [(0, None), (0, None), (0, None)]

res = opt.linprog(c=c, A\_ub=A, b\_ub=b, bounds=bounds)

print(f"Optimal Solution: {res.x}") print(f"Optimal Objective Value: {res.fun}")

output :

Optimal Solution: None Optimal Objective Value: None

**B ) Attempt any ONE of the following**

**I ) Write a python program to apply each of the following transformation on the point P=[-2,4]**

**A ) Rotation about origin through an angle 48 degree**

**B ) Scaling in X-coordinate by factor 2**

**C ) Reflection through the line y=2x-3**

**D ) Shering in X direction by 7 units**

**-**

import math P = [-2, 4]

theta = math.radians(48)

x = P[0]\*math.cos(theta) - P[1]\*math.sin(theta) y = P[0]\*math.sin(theta) + P[1]\*math.cos(theta) print(f"Rotation about origin: ({x:.2f}, {y:.2f})")

x = P[0]\*2 y = P[1]

print(f"Scaling in X-coordinate: ({x:.2f}, {y:.2f})")

m = 2

c = -3

d = (P[0] + P[1]\*m + 2\*c)/(m\*m + 1) x = 2\*d - P[0]

y = 2\*m\*d + 2\*c - P[1]

print(f"Reflection through the line y=2x-3: ({x:.2f}, {y:.2f})")

shx = 7

x = P[0] + shx\*P[1] y = P[1]

print(f"Shearing in X direction: ({x:.2f}, {y:.2f})")

output :

Rotation about origin: (-4.31, 1.19)

Scaling in X-coordinate: (-4.00, 4.00)

Reflection through the line y=2x-3: (2.00, -10.00) Shearing in X direction: (26.00, 4.00)

**II ) Find combined transformation of the line segment between the points A[4,-1] and B[3,0] for the following sequence of transformation**

**First rotation about origin through an angle πc ;followed by scaling in x coordinate by 3 units followed by reflection through the line y=x**

**-**

import matplotlib.pyplot as plt import numpy as np

t = np.linspace(0, 1, 100) x = 4 - t

y = -1 + t

plt.plot(x, y, label='Original line segment') x, y = -y, x

x \*= 3

x, y = y, x

plt.plot(x, y, label='Transformed line segment') plt.xlim(-5, 15)

plt.ylim(-5, 15) plt.xlabel('X-axis') plt.ylabel('Y-axis') plt.legend() plt.show()

