**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 4 Page.no:- Class:- BCS**

**Q1 Attempt any ONE of the following**

1. **Using Python plot the surface plot the function z=cos(x2+y2-0.5) in the intervalfrom**

**-1< x,y<1**



import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

# Create a meshgrid of x and y values x = np.linspace(-1, 1, 100)

y = np.linspace(-1, 1, 100) X, Y = np.meshgrid(x, y)

# Calculate the values of z for each (x, y) pair Z = np.cos(X\*\*2 + Y\*\*2 - 0.5)

# Create a 3D plot fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d') ax.plot\_surface(X, Y, Z)

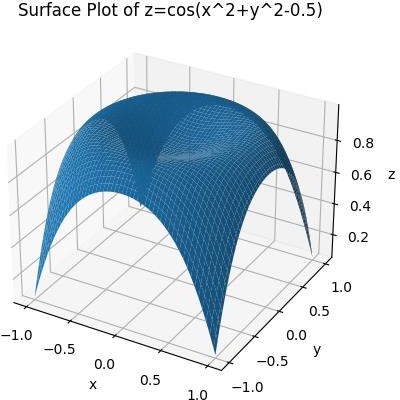
# Set the labels for the axes and the title of the plot ax.set\_xlabel('x')

ax.set\_ylabel('y') ax.set\_zlabel('z')

ax.set\_title('Surface Plot of z=cos(x^2+y^2-0.5)')

# Show the plot plt.show()

output :



# Generate 3D surface plot for the function f(x)=sin(x2+y2) in the interval [0,10]

-

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

def f(x, y):

return np.sin(x\*\*2 + y\*\*2)

x = np.linspace(0, 10, 100)

y = np.linspace(0, 10, 100) X, Y = np.meshgrid(x, y)

Z = f(X, Y)

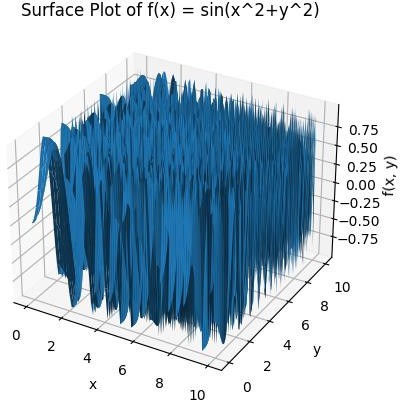
fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d') ax.plot\_surface(X, Y, Z)

ax.set\_xlabel('x') ax.set\_ylabel('y') ax.set\_zlabel('f(x, y)')

ax.set\_title('Surface Plot of f(x) = sin(x^2+y^2)') plt.show()

output :



# c) Write a python program to generate 3D plot of the function z=sin X+cos Y in the interval -10 <x ,y <10



import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D def f(x, y):

return np.sin(x) + np.cos(y)

x = np.linspace(-10, 10, 100)

y = np.linspace(-10, 10, 100) X, Y = np.meshgrid(x, y)

Z = f(X, Y)

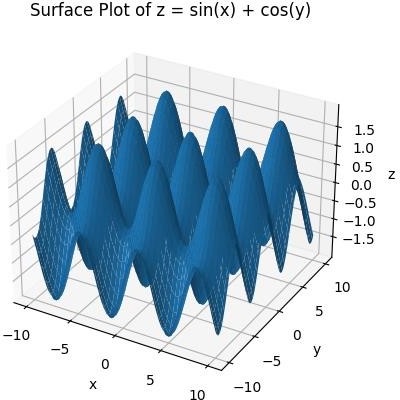
fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d') ax.plot\_surface(X, Y, Z)

ax.set\_xlabel('x') ax.set\_ylabel('y') ax.set\_zlabel('z')

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

output :



# Q 2 Attempt any TWO of the following

**A ) Using python generate triangle with vertices (0,0),(4,0),(4,3) check wheter the tringle is Right angle tringle**



import matplotlib.pyplot as plt vertices = [(0, 0), (4, 0), (4, 3)]

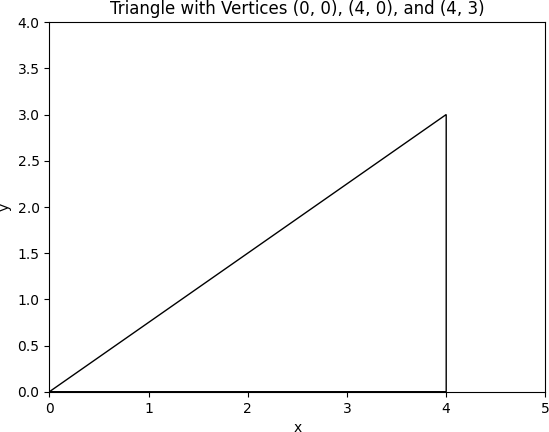
triangle = plt.Polygon(vertices, closed=True, fill=False) fig, ax = plt.subplots()

ax.add\_patch(triangle) ax.set\_xlim([0, 5])

ax.set\_ylim([0, 4]) ax.set\_xlabel('x') ax.set\_ylabel('y')

ax.set\_title('Triangle with Vertices (0, 0), (4, 0), and (4, 3)') plt.show()

output :

:

# B ) Generate vector x in interval [-7,7] using numpy package with 50 subintervals



import numpy as np n = 50

x = np.linspace(-7, 7, n) print(x)

output :

|  |  |  |
| --- | --- | --- |
| [-7. | -6.71428571 -6.42857143 -6.14285714 -5.85714286 -5.57142857 | |
| -5.28571429 -5. -4.71428571 -4.42857143 -4.14285714 | | -3.85714286 |
| -3.57142857 -3.28571429 -3. -2.71428571 -2.42857143 | | -2.14285714 |
| -1.85714286 -1.57142857 -1.28571429 -1. -0.71428571 | | -0.42857143 |
| -0.14285714 0.14285714 0.42857143 0.71428571 1. | | 1.28571429 |

1.57142857 1.85714286 2.14285714 2.42857143 2.71428571 3.

3.28571429 3.57142857 3.85714286 4.14285714 4.42857143 4.71428571

5. 5.28571429 5.57142857 5.85714286 6.14285714 6.42857143

6.71428571 7. ]

# C ) Write a python program to find the area and perimeter of the ∆ABC ,where A[0,0],B[6,0],C[4,4]



import math

A = [0, 0]

B = [6, 0]

C = [4, 4]

AB = math.sqrt((B[0]-A[0])\*\*2 + (B[1]-A[1])\*\*2)

BC = math.sqrt((C[0]-B[0])\*\*2 + (C[1]-B[1])\*\*2)

CA = math.sqrt((A[0]-C[0])\*\*2 + (A[1]-C[1])\*\*2)

perimeter = AB + BC + CA s = perimeter / 2

area = math.sqrt(s \* (s - AB) \* (s - BC) \* (s - CA)) print("Perimeter of the triangle: {:.2f}".format(perimeter)) print("Area of the triangle: {:.2f}".format(area))

output :

Perimeter of the triangle: 16.13 Area of the triangle: 12.00

**Q3 Attempt the following**

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

**I ) Write a Python program to solve the following LPP : MAX Z=5x+3y**

**Subject to x+y≤20**

**2x+5y≤1 X,y≥0**



import pulp

problem = pulp.LpProblem("Maximizing Z", pulp.LpMaximize) x = pulp.LpVariable('x', lowBound=0, cat='Continuous')

y = pulp.LpVariable('y', lowBound=0, cat='Continuous') problem += 5\*x + 3\*y, "Z"

problem += x + y <= 20, "Constraint 1" problem += 2\*x + 5\*y <= 10, "Constraint 2" status = problem.solve()

print("Solution status:", pulp.LpStatus[status])

print("Optimal solution:") print("x =", pulp.value(x))

print("y =", pulp.value(y))

print("Z =", pulp.value(problem.objective))

**II) Write a python program to display the following LPP by using pulp module and simplex method Find its optimal solution if exist**

**Max Z=4x+y+3z+5w Subject to 4x+6y-5z-4w≥20**

**-3x-2y+4z+w≤10**

**-8x-3y+3z+2w≤20 X+y≤11 X,y,z,w≥0**



import pulp

problem = pulp.LpProblem("Maximizing Z", pulp.LpMaximize) x = pulp.LpVariable('x', lowBound=0, cat='Continuous')

y = pulp.LpVariable('y', lowBound=0, cat='Continuous')

z = pulp.LpVariable('z', lowBound=0, cat='Continuous') w = pulp.LpVariable('w', lowBound=0, cat='Continuous')

problem += 4\*x + y + 3\*z + 5\*w, "Z"

problem += 4\*x + 6\*y - 5\*z - 4\*w >= 20, "Constraint 1" problem += -3\*x - 2\*y + 4\*z + w <= 10, "Constraint 2" problem += -8\*x - 3\*y + 3\*z + 2\*w <= 20, "Constraint 3" problem += x + y <= 11, "Constraint 4"

status = problem.solve()

print("Solution status:", pulp.LpStatus[status]) print("Optimal solution:")

print("x =", pulp.value(x))

print("y =", pulp.value(y))

print("z =", pulp.value(z))

print("w =", pulp.value(w))

print("Z =", pulp.value(problem.objective))

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

**I ) Apply Python program in each of the following transformation on the point P[3,8]**

**A ) Reflection through X- axis**

**B ) Scaling in X- coordinate by factor 6**

**C ) Rotation about origin thr ough an angel 30⸰**

**D ) Reflection through the line y=-x**



P = [3, 8]

P\_reflected = [P[0], -P[1]] print("Reflected point:", P\_reflected)

P = [3, 8]

P\_scaled = [6 \* P[0], P[1]] print("Scaled point:", P\_scaled)

import math P = [3, 8]

angle = math.radians(30)

P\_rotated = [P[0] \* math.cos(angle) - P[1] \* math.sin(angle), P[0] \* math.sin(angle) + P[1] \* math.cos(angle)]

print("Rotated point:", P\_rotated)

P = [3, 8]

P\_reflected = [(P[1] + P[0]) / 2, (P[1] + P[0]) / 2]

print("Reflected point:", P\_reflected)

output :

Reflected point: [3, -8]

Scaled point: [18, 8]

Rotated point: [-1.4019237886466835, 8.428203230275509]

Reflected point: [5.5, 5.5]

**II) Write a Python program to plot 2D x-axis and Y-axis in black color,in the same diagram plot**

**A )Green triangle with vertices [5,4],[7,4],[6,6]**

**B ) Blue rectangle with vertices [2,2],[10,2],[10,8],[2,8]**

1. **Red polygon with vertices [6,2],[10,4],[8,7],[2,4]**
2. **Isosceles triangle with vertices [0,0],[4,0],[2,4]**



import matplotlib.pyplot as plt

triangle = plt.Polygon([[5, 4], [7, 4], [6, 6]], color='green') plt.gca().add\_patch(triangle)

rectangle = plt.Polygon([[2, 2], [10, 2], [10, 8], [2, 8]], color='blue') plt.gca().add\_patch(rectangle)

polygon = plt.Polygon([[6, 2], [10, 4], [8, 7], [2, 4]], color='red') plt.gca().add\_patch(polygon)

triangle = plt.Polygon([[0, 0], [4, 0], [2, 4]], color='black') plt.gca().add\_patch(triangle)

plt.axhline(0, color='black') plt.axvline(0, color='black') plt.xlim(-2, 12)

plt.ylim(-2, 10) plt.show()

