AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
International Credit Hours Programs (ICHEP)
Mechatronics and Automation

Computational Intelligence (CSE473)

Lab 3 Report

Name	ID	Section
Mahmoud Elsayd Abdelqader Labib Eldwakhly	21P0017	1

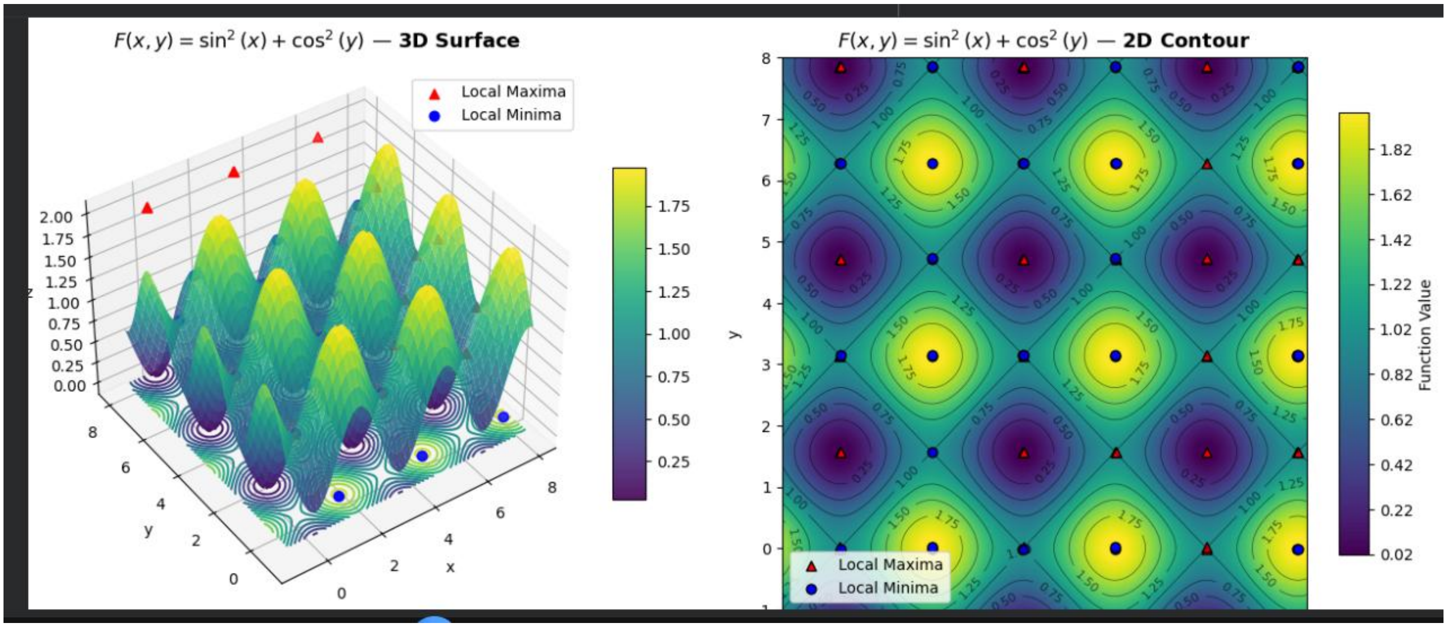
Submitted to Dr Hossam Hassan & Eng. Ali Ahmed

Fall 2025

```python
# ════════════════════════════════════════════════════════════════
# CSE473: Computational Intelligence
# Lab Assignment #03
# Name: Mahmoud Elsayd Eldwakhly
# ID : 21P0017
# ════════════════════════════════════════════════════════════════


import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# ──────────────────────────────────────────────────────────────────────────
# GRID AND FUNCTIONS
# ──────────────────────────────────────────────────────────────────────────
x = np.linspace(-1, 8, 400)
y = np.linspace(-1, 8, 400)
X, Y = np.meshgrid(x, y)

F = np.sin(X)**2 + np.cos(Y)**2

G1 = 9 / np.sqrt((X - 2)**2 + (Y - 4)**2 + 1e-6) \
   + 12 / np.sqrt((X - 5)**2 + (Y - 2)**2 + 1e-6) \
   + 25 / np.sqrt((X - 4)**2 + (Y - 5)**2 + 1e-6)

G2 = np.sqrt((X - 2)**2 + (Y - 4)**2) / 9 \
   + np.sqrt((X - 5)**2 + (Y - 2)**2) / 12 \
   + np.sqrt((X - 4)**2 + (Y - 5)**2) / 25

# ──────────────────────────────────────────────────────────────────────────
# FIND EXTREMA (NO SCIPY)
# ──────────────────────────────────────────────────────────────────────────
def find_extrema(Z):
    gx, gy = np.gradient(Z)
    grad_mag = np.sqrt(gx**2 + gy**2)
    candidates = np.where(grad_mag < np.percentile(grad_mag, 0.05))
    gxx, _ = np.gradient(gx)
    _, gyy = np.gradient(gy)
    lap = gxx + gyy
    maxima = (lap < 0)
    minima = (lap > 0)
    max_pts = [(x[i], y[j], Z[i, j]) for i, j in zip(candidates[0], candidates[1]) if maxima[i, j]]
    min_pts = [(x[i], y[j], Z[i, j]) for i, j in zip(candidates[0], candidates[1]) if minima[i, j]]
    return np.array(max_pts), np.array(min_pts)

# ──────────────────────────────────────────────────────────────────────────
# COMBINED PLOT FUNCTION (3D + 2D)
# ──────────────────────────────────────────────────────────────────────────
def plot_function(X, Y, Z, title, cmap='viridis', clip_percent=1):
    # Clip for visual clarity
    vmin, vmax = np.percentile(Z, clip_percent), np.percentile(Z, 100 - clip_percent)
    Z_plot = np.clip(Z, vmin, vmax)

    # Compute extrema
    max_pts, min_pts = find_extrema(Z)

    fig = plt.figure(figsize=(13,6))

    # ──────────────────── 3D SURFACE ────────────────────
    ax3d = fig.add_subplot(1, 2, 1, projection='3d')
    surf = ax3d.plot_surface(X, Y, Z_plot, cmap=cmap, linewidth=0, antialiased=True, alpha=0.9)
    ax3d.contour(X, Y, Z_plot, 15, offset=np.min(Z_plot), cmap=cmap)

    # Plot extrema
    if len(max_pts) > 0:
        ax3d.scatter(max_pts[:,0], max_pts[:,1], max_pts[:,2],
                     color='red', s=40, marker='^', label='Local Maxima')
    if len(min_pts) > 0:
        ax3d.scatter(min_pts[:,0], min_pts[:,1], min_pts[:,2],
                     color='blue', s=40, marker='o', label='Local Minima')

    ax3d.set_title(title + " – 3D Surface", fontsize=13, weight='bold', pad=15)
    ax3d.set_xlabel('x')
    ax3d.set_ylabel('y')
    ax3d.set_zlabel('z')
    ax3d.view_init(elev=40, azim=235)
    ax3d.legend()
    fig.colorbar(surf, ax=ax3d, shrink=0.6, aspect=10)

    # ──────────────────── 2D CONTOUR ────────────────────
    ax2d = fig.add_subplot(1, 2, 2)
    cf = ax2d.contourf(X, Y, Z_plot, levels=100, cmap=cmap)
    cs = ax2d.contour(X, Y, Z_plot, colors='black', linewidths=0.5, alpha=0.6)
    ax2d.clabel(cs, inline=True, fontsize=8, fmt="%.2f")
    fig.colorbar(cf, ax=ax2d, shrink=0.8, aspect=15, label='Function Value')

    # Plot extrema
    if len(max_pts) > 0:
        ax2d.scatter(max_pts[:,0], max_pts[:,1], color='red', s=40, marker='^', edgecolor='black', label='Local Maxima')
    if len(min_pts) > 0:
        ax2d.scatter(min_pts[:,0], min_pts[:,1], color='blue', s=40, marker='o', edgecolor='black', label='Local Minima')

    ax2d.set_title(title + " – 2D Contour", fontsize=13, weight='bold')
    ax2d.set_xlabel('x')
    ax2d.set_ylabel('y')
    ax2d.legend()

    plt.tight_layout()
    plt.show()
```

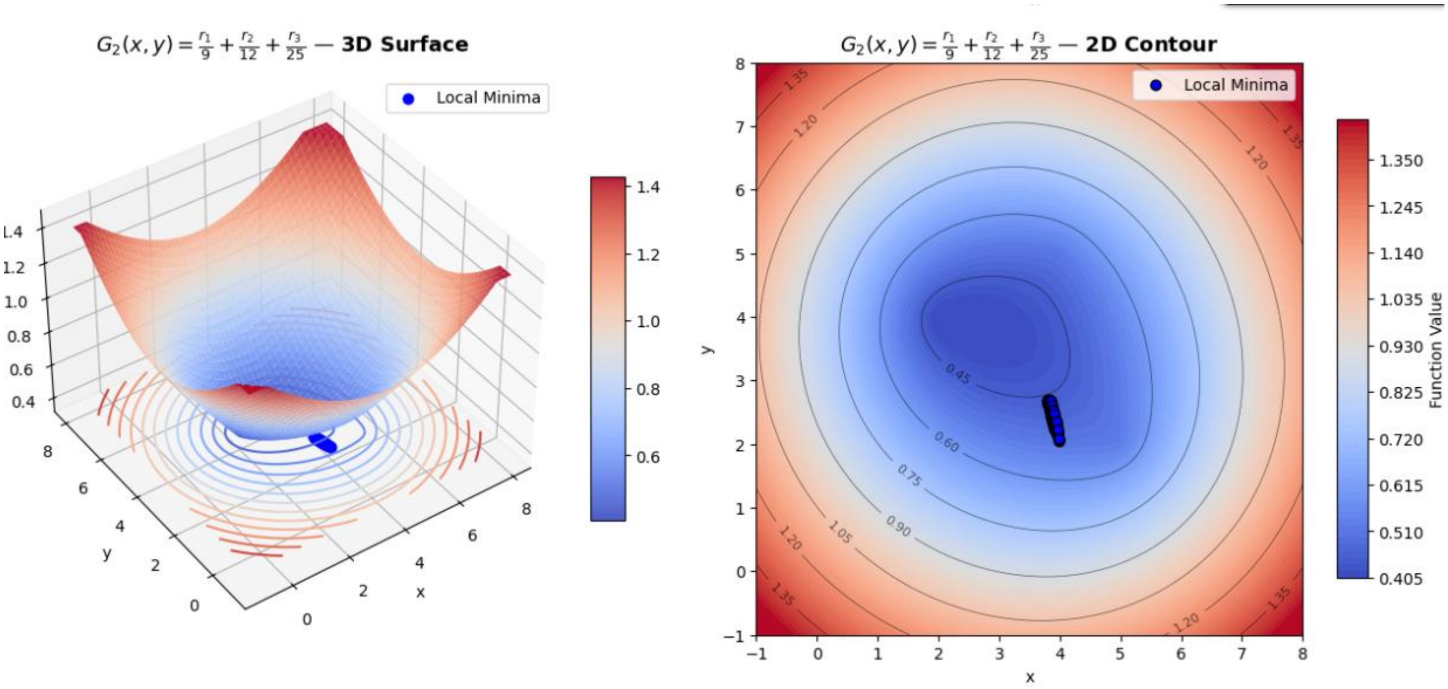
```python

# ──────────────────────────────────────────────────────────────────────────────
# PLOT ALL THREE FUNCTIONS
# ──────────────────────────────────────────────────────────────────────────────
plot_function(X, Y, F, r"$F(x,y) = \sin^2(x) + \cos^2(y)$", cmap='viridis')
plot_function(X, Y, G1, r"$G_1(x,y) = \frac{9}{r_1} + \frac{12}{r_2} + \frac{25}{r_3}$", cmap='inferno', clip_percent=5)
plot_function(X, Y, G2, r"$G_2(x,y) = \frac{r_1}{9} + \frac{r_2}{12} + \frac{r_3}{25}$", cmap='coolwarm')

```

Output:



$F(x,y) = \sin^2(x) + \cos^2(y)$ — **3D Surface**

$F(x,y) = \sin^2(x) + \cos^2(y)$ — **2D Contour**



$G_2(x,y) = \frac{r_1}{9} + \frac{r_2}{12} + \frac{r_3}{25}$ — **3D Surface**

$G_2(x,y) = \frac{r_1}{9} + \frac{r_2}{12} + \frac{r_3}{25}$ — **2D Contour**