

Multi-Variable Calculus

|  |  |
| --- | --- |
| Assignment 2 | Code Implementation of Functions |
| Name | Muhammad Abdullah |
| Sap ID | 53457 |
| Section | BSCS-3B |
| Instructor | Dr. Maria Akram |

To plot our function we will use python libraries of matplotlib for plotting graph of functions and numpy for the case of mathematical calculation and computations. Before diving into the code we need to first make sure that our environment is setup correctly with all dependencies installed and configured on our machines. Firstly install python from the following link on your local machine:

<https://www.python.org/downloads/>

I have windows on my machine you can download for any kind of operating system you have on this website.

After downloading the python we need to ensure that we have our concerning libraries installed i.e. numpy & python. We will use pip Command to install these dependencies. The commands are as follow,

The command for Numpy: “pip install numpy”

The command for matpotlib: “pip install matplotlib”

Now open jupyter notebook by entering the following command in command prompt

The command for opening jupyter notebook: “jupyter notebook”

Now import libraries with following commands

Commands:

import numpy as np

import matplotlib.pyplot as plt

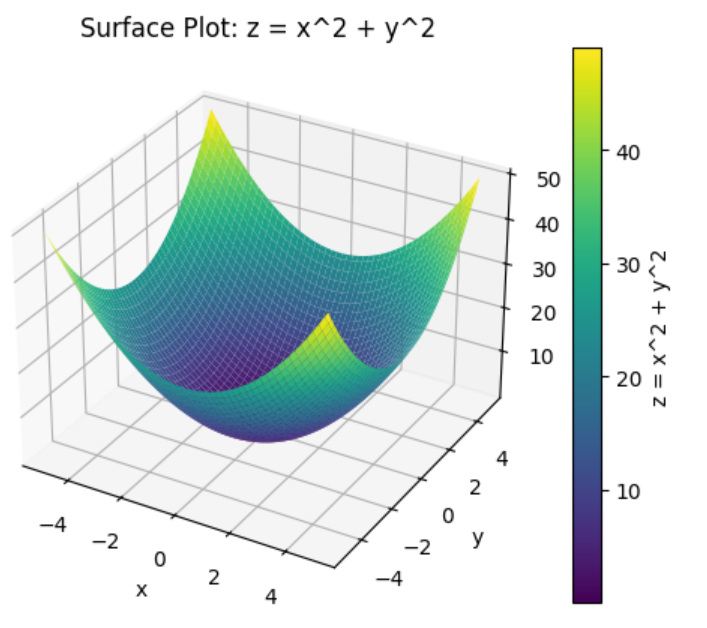
from mpl\_toolkits.mplot3d import Axes3D # For 3d plotting

1. **Function 1:**

Code for function 1 is as follows in python:

|  |
| --- |
| def plot\_function\_1():  x = np.linspace(-5, 5, 100)  y = np.linspace(-5, 5, 100)  x, y = np.meshgrid(x, y)  z = x\*\*2 + y\*\*2  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  surf = ax.plot\_surface(x, y, z, cmap='viridis')  fig.colorbar(surf, label="z = x^2 + y^2")  ax.set\_title("Surface Plot: z = x^2 + y^2")  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')  plt.show()  plt.savefig('Function1.png')  plot\_function\_1() |

Figure:

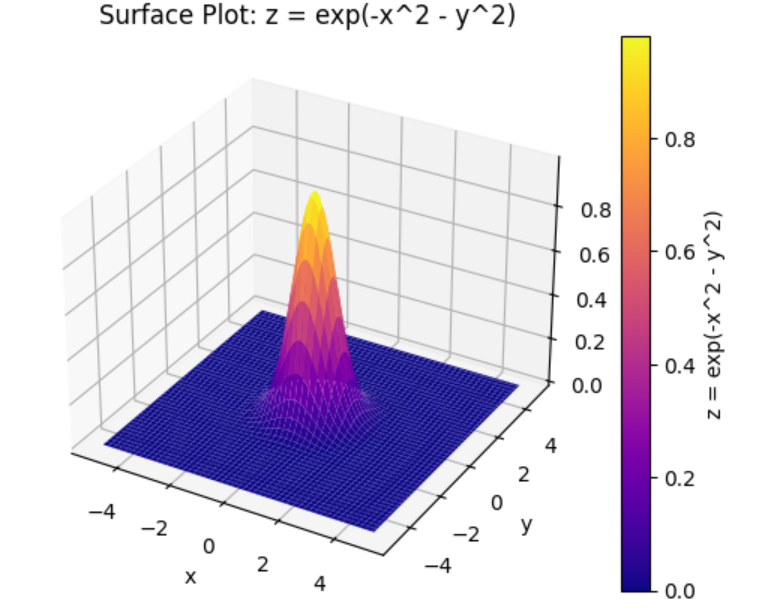


1. **Function 2:**

Code for function 2 is as follows in python:

|  |
| --- |
| def plot\_function\_2():  x = np.linspace(-5, 5, 100)  y = np.linspace(-5, 5, 100)  x, y = np.meshgrid(x, y)  z = np.exp(-x\*\*2 - y\*\*2)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  surf = ax.plot\_surface(x, y, z, cmap='plasma')  fig.colorbar(surf, label="z = exp(-x^2 - y^2)")  ax.set\_title("Surface Plot: z = exp(-x^2 - y^2)")  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')  plt.show()  plt.savefig('Function2.png')  plot\_function\_2() |

Figure:

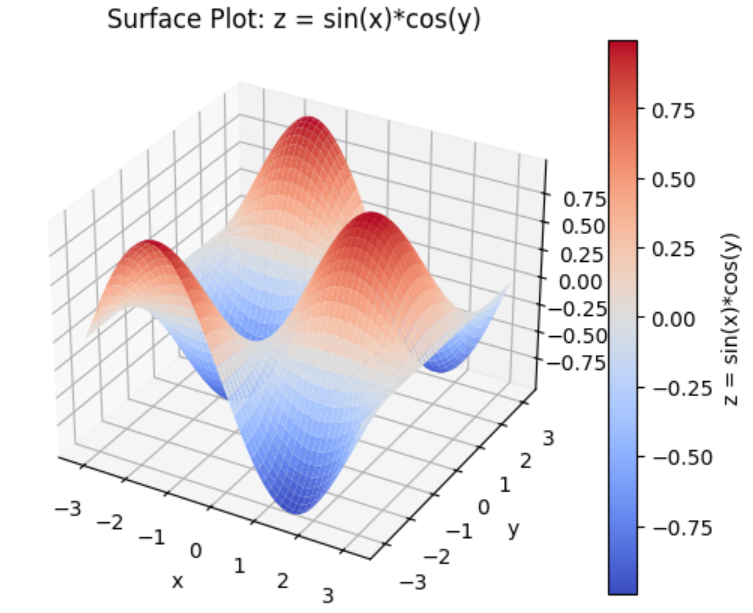


1. **Function 3:**

Code for function 3 is as follows in python:

|  |
| --- |
| def plot\_function\_3():  x = np.linspace(-np.pi, np.pi, 100)  y = np.linspace(-np.pi, np.pi, 100)  x, y = np.meshgrid(x, y)  z = np.sin(x) \* np.cos(y)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  surf = ax.plot\_surface(x, y, z, cmap='coolwarm')  fig.colorbar(surf, label="z = sin(x)\*cos(y)")  ax.set\_title("Surface Plot: z = sin(x)\*cos(y)")  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')  plt.show()  plt.savefig('Function3.png')  plot\_function\_3() |

Figure:

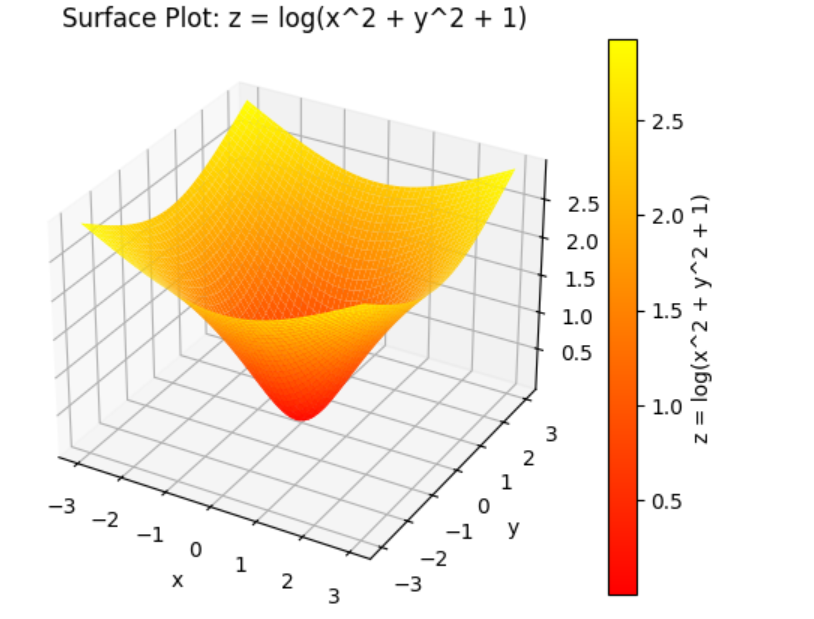


1. **Function 4:**

Code for function 4 is as follows in python:

|  |
| --- |
| def plot\_function\_4():  x = np.linspace(-3, 3, 100)  y = np.linspace(-3, 3, 100)  x, y = np.meshgrid(x, y)  z = np.log(x\*\*2 + y\*\*2 + 1)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  surf = ax.plot\_surface(x, y, z, cmap='autumn')  fig.colorbar(surf, label="z = log(x^2 + y^2 + 1)")  ax.set\_title("Surface Plot: z = log(x^2 + y^2 + 1)")  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')  plt.show()  plt.savefig('Function4.png')  plot\_function\_4() |

Figure:

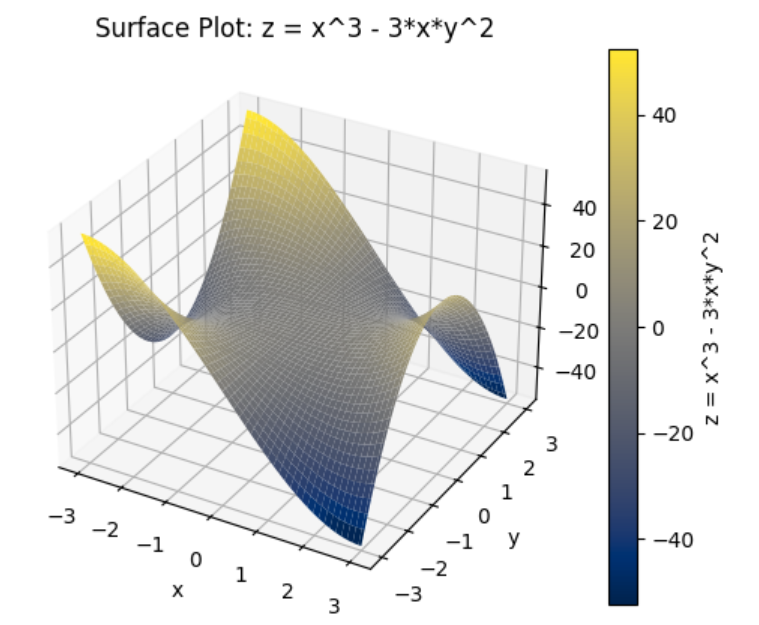


1. **Function 5:**

Code for function 5 is as follows in python:

|  |
| --- |
| def plot\_function\_5():  x = np.linspace(-3, 3, 100)  y = np.linspace(-3, 3, 100)  x, y = np.meshgrid(x, y)  z = x\*\*3 - 3\*x\*y\*\*2  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  surf = ax.plot\_surface(x, y, z, cmap='cividis')  fig.colorbar(surf, label="z = x^3 - 3\*x\*y^2")  ax.set\_title("Surface Plot: z = x^3 - 3\*x\*y^2")  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')  plt.show()  plt.savefig('Function5.png')  plot\_function\_5() |

Figure:

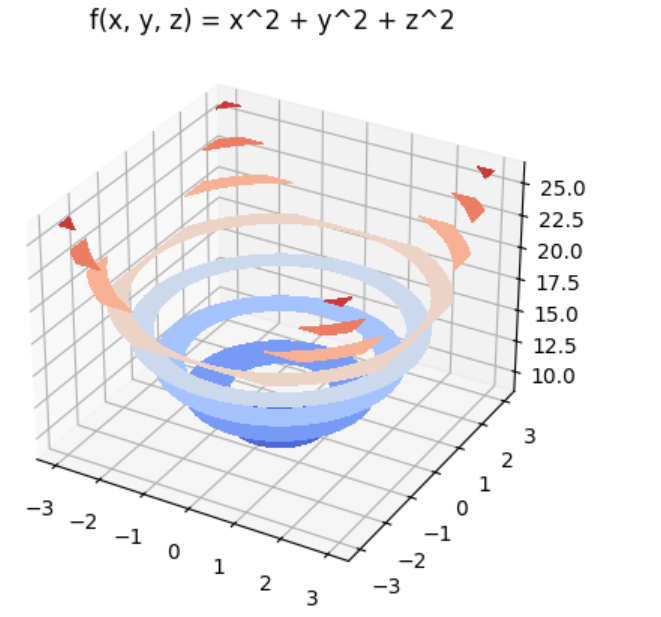


1. **Function 6:**

Code for function 6 is as follows in python:

|  |
| --- |
| def plot\_3d\_slice(f, title, domain=(-3, 3)):  x = np.linspace(domain[0], domain[1], 30)  y = np.linspace(domain[0], domain[1], 30)  z = np.linspace(domain[0], domain[1], 30)  x, y, z = np.meshgrid(x, y, z)  values = f(x, y, z)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  slice\_positions = [0, 0, 0] # Slicing at x=0, y=0, z=0  ax.contourf(x[:, :, slice\_positions[2]],  y[:, :, slice\_positions[2]],  values[:, :, slice\_positions[2]], cmap='coolwarm')  ax.set\_title(title)  plt.show()  plt.savefig('Function6.png')  def f6(x, y, z):  return x\*\*2 + y\*\*2 + z\*\*2  plot\_3d\_slice(f6, "f(x, y, z) = x^2 + y^2 + z^2") |

Figure:

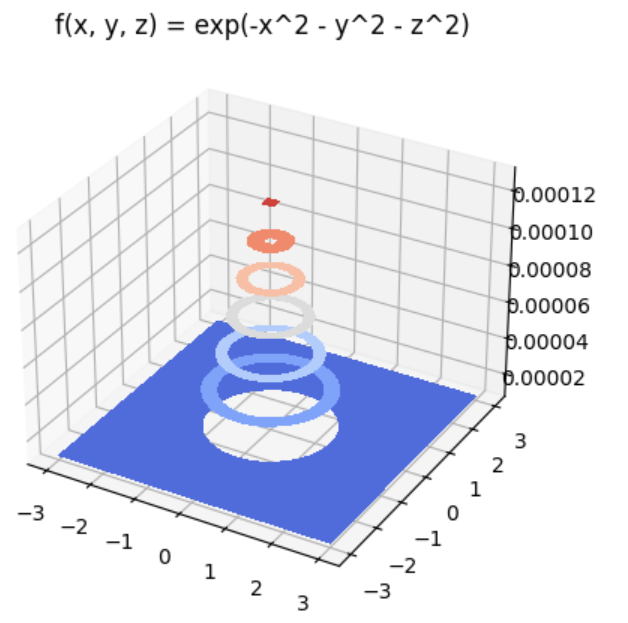


1. **Function 7:**

Code for function 7 is as follows in python:

|  |
| --- |
| def plot\_3d\_slice(f, title, domain=(-3, 3)):  x = np.linspace(domain[0], domain[1], 30)  y = np.linspace(domain[0], domain[1], 30)  z = np.linspace(domain[0], domain[1], 30)  x, y, z = np.meshgrid(x, y, z)  values = f(x, y, z)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  slice\_positions = [0, 0, 0] # Slicing at x=0, y=0, z=0  ax.contourf(x[:, :, slice\_positions[2]],  y[:, :, slice\_positions[2]],  values[:, :, slice\_positions[2]], cmap='coolwarm')  ax.set\_title(title)  plt.show()  plt.savefig('Function7.png')  def f7(x, y, z):  return np.exp(-x\*\*2 - y\*\*2 - z\*\*2)  plot\_3d\_slice(f7, "f(x, y, z) = exp(-x^2 - y^2 - z^2)") |

Figure:

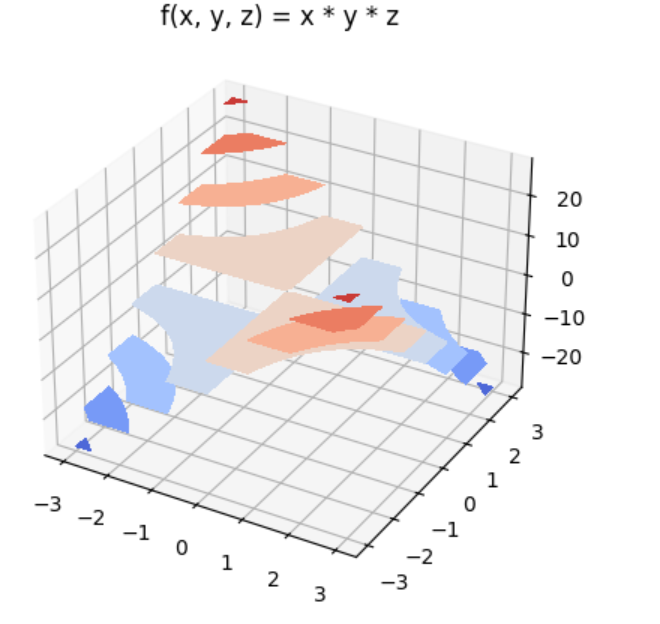


1. **Function 8:**

Code for function 8 is as follows in python:

|  |
| --- |
| def plot\_3d\_slice(f, title, domain=(-3, 3)):  x = np.linspace(domain[0], domain[1], 30)  y = np.linspace(domain[0], domain[1], 30)  z = np.linspace(domain[0], domain[1], 30)  x, y, z = np.meshgrid(x, y, z)  values = f(x, y, z)  fig = plt.figure()  ax = fig.add\_subplot(111, projection='3d')  slice\_positions = [0, 0, 0] # Slicing at x=0, y=0, z=0  ax.contourf(x[:, :, slice\_positions[2]],  y[:, :, slice\_positions[2]],  values[:, :, slice\_positions[2]], cmap='coolwarm')  ax.set\_title(title)  plt.show()  plt.savefig('Function8.png')  def f8(x, y, z):  return x \* y \* z  plot\_3d\_slice(f8, "f(x, y, z) = x \* y \* z") |

Figure:

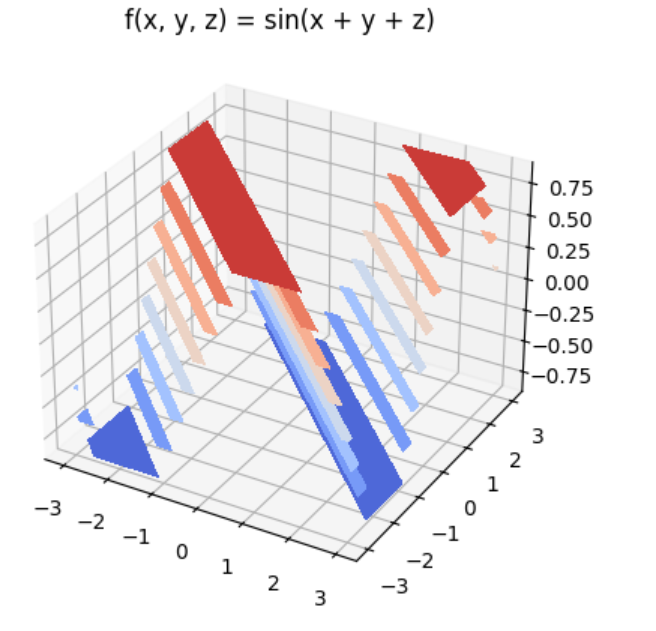


1. **Function 9:**

Code for function 9 is as follows in python:

|  |
| --- |
| def plot\_3d\_slice(f, title, domain=(-3, 3)):      x = np.linspace(domain[0], domain[1], 30)      y = np.linspace(domain[0], domain[1], 30)      z = np.linspace(domain[0], domain[1], 30)      x, y, z = np.meshgrid(x, y, z)      values = f(x, y, z)      fig = plt.figure()      ax = fig.add\_subplot(111, projection='3d')      slice\_positions = [0, 0, 0]  # Slicing at x=0, y=0, z=0      ax.contourf(x[:, :, slice\_positions[2]],                  y[:, :, slice\_positions[2]],                  values[:, :, slice\_positions[2]], cmap='coolwarm')      ax.set\_title(title)      plt.show()      plt.savefig('Function9.png')  def f9(x, y, z):  return np.sin(x + y + z)  plot\_3d\_slice(f9, "f(x, y, z) = sin(x + y + z)") |

Figure:



1. **Function 10:**

Code for function 10 is as follows in python:

|  |
| --- |
| def plot\_3d\_slice(f, title, domain=(-3, 3)):      x = np.linspace(domain[0], domain[1], 30)      y = np.linspace(domain[0], domain[1], 30)      z = np.linspace(domain[0], domain[1], 30)      x, y, z = np.meshgrid(x, y, z)      values = f(x, y, z)      fig = plt.figure()      ax = fig.add\_subplot(111, projection='3d')      slice\_positions = [0, 0, 0]  # Slicing at x=0, y=0, z=0      ax.contourf(x[:, :, slice\_positions[2]],                  y[:, :, slice\_positions[2]],                  values[:, :, slice\_positions[2]], cmap='coolwarm')      ax.set\_title(title)      plt.show()      plt.savefig('Function7.png')  def f10(x, y, z):  return np.sqrt(x\*\*2 + y\*\*2 + z\*\*2)  plot\_3d\_slice(f10, "f(x, y, z) = sqrt(x^2 + y^2 + z^2)") |

Figure:

