1.

b)

import numpy as np

import matplotlib.pyplot as plt

def func(x,y):

    return 100\*(y-x\*\*2)\*\*2 + (1-x)\*\*2

# create two one-dimensional grids using linspace

x = np.linspace(-5, 5, 50)

y = np.linspace(-5, 5, 50)

# combine the two one-dimensional grids into one two-dimensional grid

X, Y = np.meshgrid(x,y)

# evaluate the function at each element of the two-dimensional grid

Z = func(X, Y)

# create plot

fig = plt.figure(figsize=(7,7))

ax = plt.axes(projection='3d')

ax.plot\_surface(X, Y, Z, cmap='viridis')

plt.show()

Chart, surface chart

Description automatically generated

c)

k=0, [-1.2 1. ]

k=1, [-1.1752809 1.38067416]

k=2, [ 0.76311487 -3.17503385]

k=3, [0.76342968 0.58282478]

k=4, [0.99999531 0.94402732]

k=5, [0.9999957 0.99999139]

k=6, [1. 1.]

k=7, [1. 1.]

import numpy as np

start = np.array([-1.2,1])

v = start

def nab(x, y):

    return np.array([-400\*(y-(x\*\*2))\*x - 2\*(1-x), 200\*(y-x\*\*2)])

def hes(x, y):

    return np.array([[1200\*(x\*\*2)-400\*y+2, -400\*x], [-400\*x, 200]])

new\_nab = nab(v[0], v[1])

print(f'k=0, {v}')

iteration = 1

while not np.linalg.norm(new\_nab, ord=2) <= 10\*\*(-6):

    new\_nab = nab(v[0], v[1])

    new\_hes = hes(v[0], v[1])

    v = v - np.linalg.inv(new\_hes) @ new\_nab

    print(f'k={iteration}, {v}')

    iteration += 1