Gradient Descent Optimization

M.Saketh Sai Ram - EE22B022

Import the necessary libraries such as `numpy` and `matplotlib` to implement gradient descent optimization and generate animation. The Python script `gradient_descent.py` contains the code that implements the optimization. The Python script contains a function `gradient_descent_optimization()`, which implements the gradient descent based optimization and gives the animation showing the convergence.

Description of gradient_descent_optimization():

- It takes a function and its gradient/s, Starting point, Learning Rate(it controls the step size during each iteration of optimization), Number of iterations, and Time interval as its arguments.
- First, it checks whether the given polynomial is 1D or 2D, and then executes the corresponding algorithm.
- It creates empty lists to store the points on the trajectory while converging.
- Then it keeps on updating the points in the trajectory based on the equations;
  - $x_{i+1} = x_i - lr \times \left(\frac{df(x)}{dx}\right)_{x=x_i}$ for 1D.
  - $x_{i+1} = x_i - lr \times \left(\frac{df(x,y)}{dx}\right)_{(x,y)=(x_i,y_i)}$ for 2D.
  - $y_{i+1} = y_i - lr \times \left(\frac{df(x,y)}{dy}\right)_{(x,y)=(x_i,y_i)}$ for 2D.
- As the gradient decreases, the values converge to a point called Minimum.
- Finally, it shows the animation and outputs the optimized values for (x, y, z) or (x, y) for the specified parameters.

1 Problem 1

1.1 Algorithm

$$\text{1D Polynomial : } f1(x) = x^2 + 3x + 8.$$

There exists only one point of minimum for this polynomial, so it can be optimized by taking the starting point anywhere in the specified range except the point of minimum. Assuming the starting point to be $(4.5, f1(4.5))$, the trajectory of optimization will be updated from this point.

1.1.1 Trajectory Parameters

- Learning Rate = 0.1
- Number of Iterations = 100
- Time Interval = 50

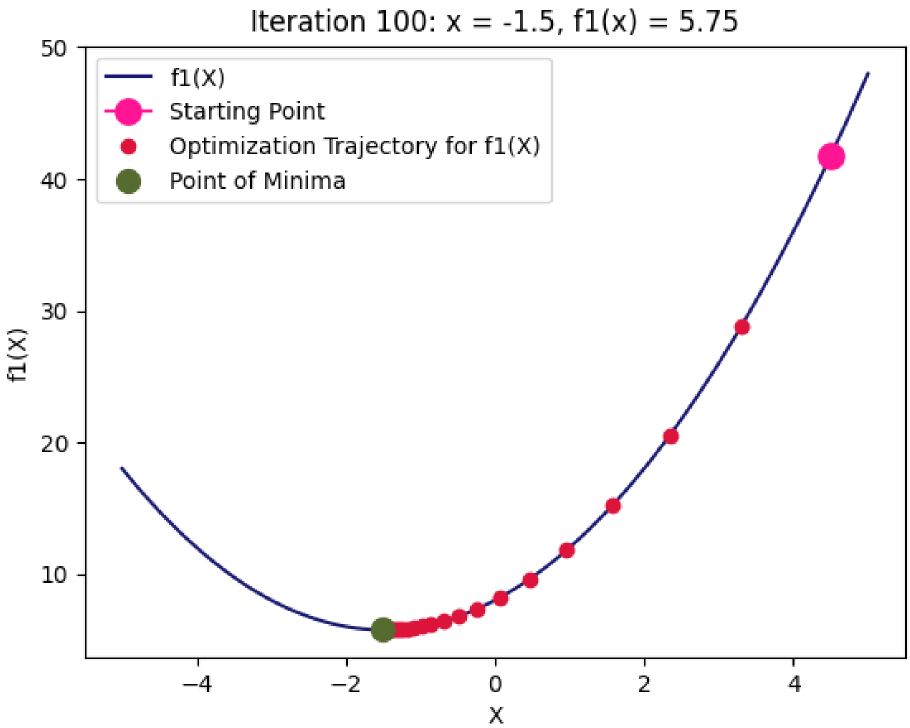
1.2 Optimized Values

The optimized values after 100 iterations are as follows,

- x = -1.499996986991167
- f1(x) = 5.750000000009079

The actual point of minimum is $(-1.5, -5.75)$, this shows that optimized values are very much closer to the actual values. Improvement in the values can be seen by increasing the number of iterations.

1.3 Plotting



2 Problem 2

2.1 Algorithm

$$\text{2D Polynomial}: \ f3(x, y) = x^4 - 16x^3 + 96x^2 - 256x + y^2 - 4y + 262.$$

By plotting the 3D graph, it is observed that it has only one point of minimum in the specified range. So, it can be optimized by taking the starting point anywhere in the specified range except the point of minimum. Assuming the starting point to be $(5, 5, f3(5, 5))$, the trajectory of optimization will be updated from this point.

2.1.1 Trajectory Parameters

- Learning Rate = 0.1
- Number of Iterations = 150
- Time Interval = 25

2.2 Optimized Values

The optimized values after 150 iterations are as follows,

- x = 4.089390037598209, y = 2.0000000000000067
- f3(x,y) = 2.0000638493498855

The actual point of minimum is $(4, 2, 2)$. Improvement in the values can be seen by increasing the number of iterations or by changing the Learning Rate or Starting Point.

2.3 Plotting

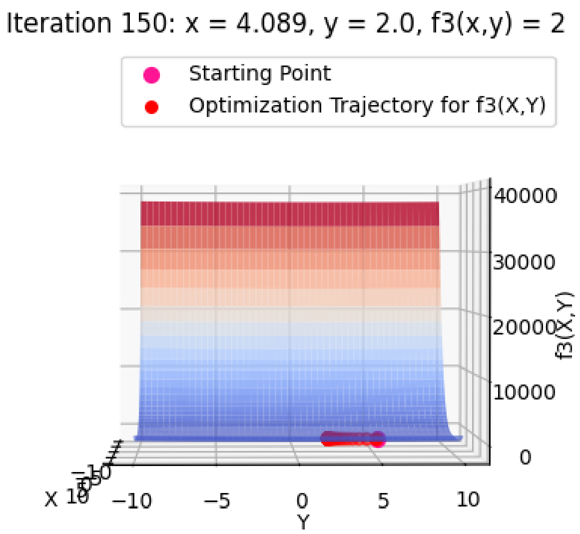


Figure 1: Starting Point

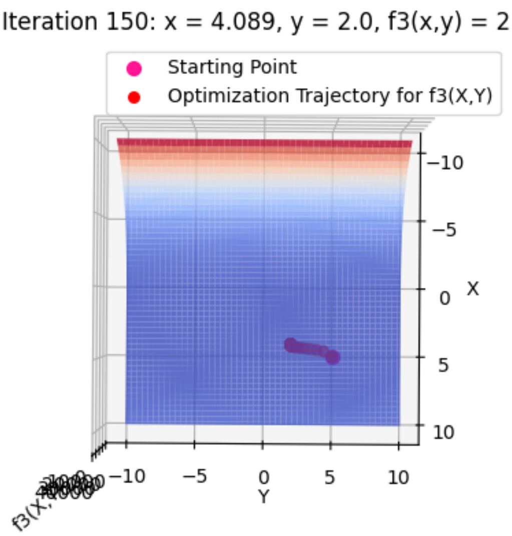
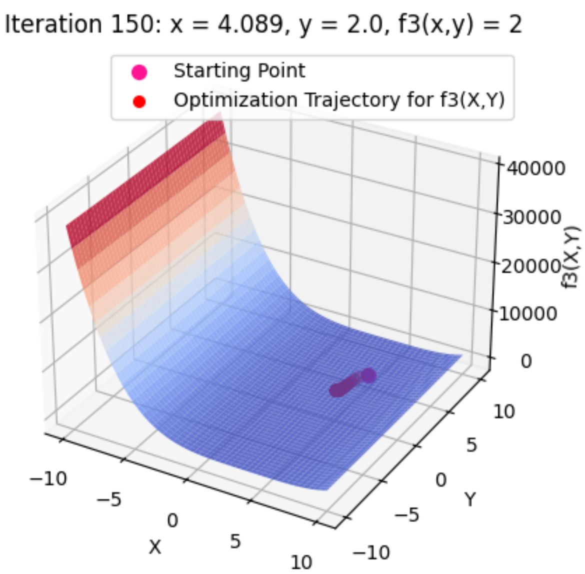


Figure 2: Trajectory



3 Problem 3

3.1 Algorithm

$$\text{2D Polynomial : } f4(x, y) = e^{-(x-y)^2} sin(y)$$

Plotting the 3D graph shows that it has one minimum and one maximum in the specified range and is an odd function with respect to x and y, i.e., $f4(-x,-y) = -f4(x,y)$. It is also observed that the minimum point lies in the region $R = \{(x,y,z) : x < 0, y < 0, z < 0\}$. Therefore, the starting point should lie in R. Here the starting point is assumed to be $(-0.1, -0.1, f4(-0.1, -0.1))$.

3.1.1 Trajectory Parameters

- Learning Rate = 0.1
- Number of Iterations = 150
- Time Interval = 25

3.2 Optimized Values

The optimized values after 150 iterations are as follows,

- x = -1.569509964441952, y = -1.5698226562516964
- f4(x,y) = -0.9999994282067574

The actual point of minimum is $(-\frac{\pi}{2}, -\frac{\pi}{2}, -1)$. Improvement in the values can be seen by increasing the number of iterations.

3.3 Plotting

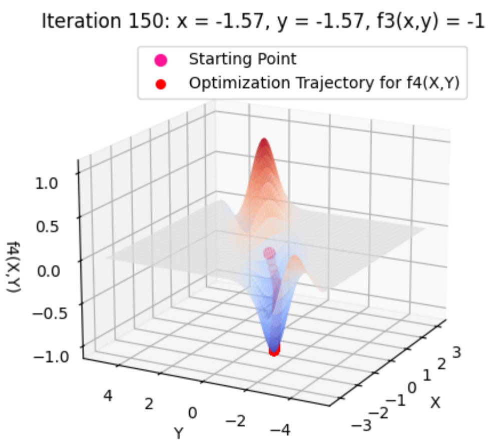


Figure 3: Starting Point

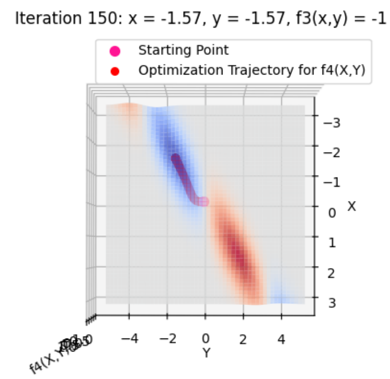
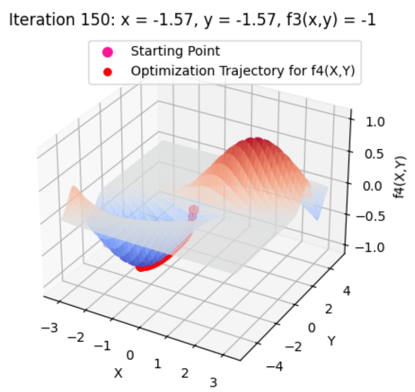


Figure 4: Trajectory



4 Problem 4

4.1 Algorithm

$$\text{1D Polynomial}: \ f5(x) = \cos^4 x - \sin^3 x - 4\sin^2 x + 4\cos x + 1.$$

By plotting the graph, it is observed that it has two local minima's and one local maxima, so the starting point is assumed to be $(0.2, f5(0.2))$(which lies on the left side of the lowest minima) to optimize it to the lowest minimum in the specified range. Finding the lowest minimum can also achieved by choosing the starting point which is just left to the maxima.

4.1.1 Trajectory Parameters

- Learning Rate = 0.05
- Number of Iterations = 100
- Time Interval = 50

4.2 Optimized values

The optimized values after 100 iterations are as follows,

- x = 1.661660812043789
- f5(x) = -4.045412051572552

The actual point of minimum is $(1.662, -4.05)$. Improvement in the values can be seen by increasing the number of iterations.

4.3 Plotting