Assignment 5

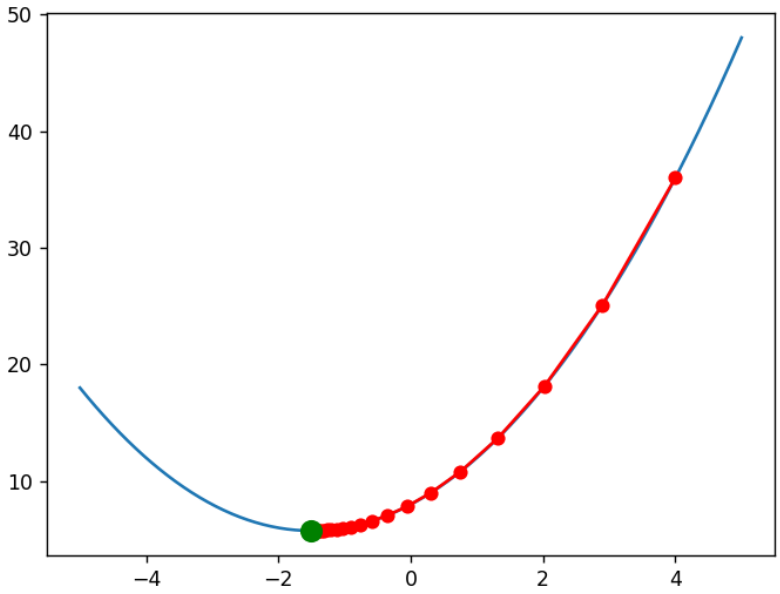
The code contains two functions, an optimizer for 1 variable, and an optimizer for 2 variables. To run the code, uncomment the respective calls, along with the respective plots. For example, while calling `optimize1d`, uncomment the line `fig, ax = plt.subplots()`, and while calling `optimize2d`, uncomment the two lines above the function call.

1 Dimension

A generic function `optimize1d` has been created, which takes in two functions, namely the function to be optimized and its derivative. Along with these two, it also takes in the range of x to search for the optimum in, and an initial guess. It uses gradient descent for optimization. Thus, before running the code, the code must be modified to create a function call to `optimize1d`, after passing in the appropriate functions. The function outputs an animation of the optimization process, and prints out the optimum value.

Function f1

After passing in function `f1` to the generic function `optimize1d`, we get the final graph as shown below:

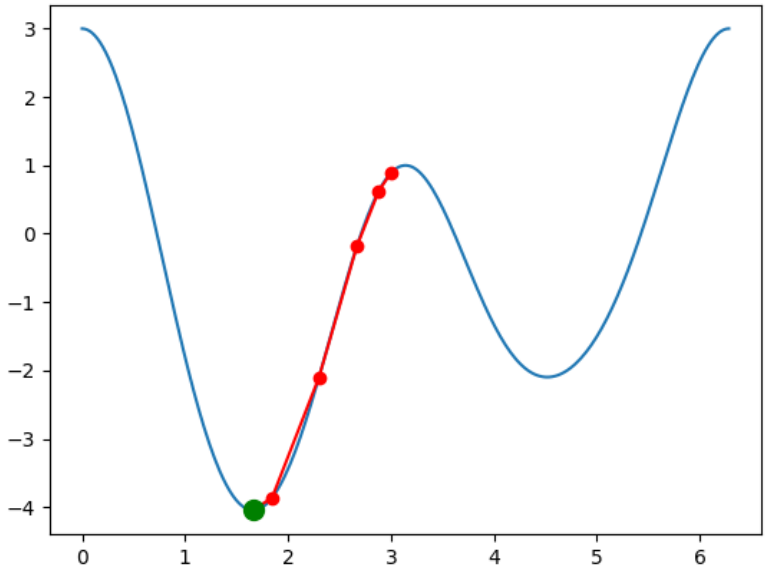


And the coordinates of the optimized point to be `(-1.499999999103704, 5.75)`. The learning rate used here was *0.1*, and the number of iterations were

100. Initial point taken was $x = 4$.

Function f5

After passing in function `f5` to the generic function `optimize1d`, we get the final graph as shown below:



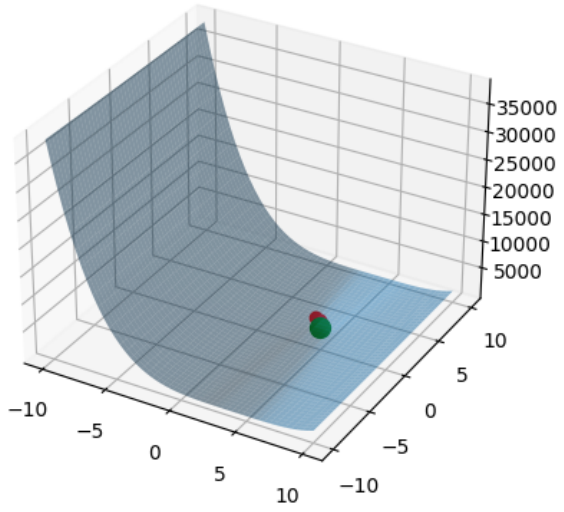
And the coordinates of the optimized point to be `(1.661865650350945 -4.045411821419813)`. The learning rate used here was *0.1*, and the number of iterations were *100*. Initial point taken was $x = 3$

2 Dimension

A generic function `optimize2d` has been created, which takes in three functions, namely the function to be optimized and its partial derivatives along x and y. Along with these two, it also takes in the range of x and y to search for the optimum in, and an initial guess. It uses gradient descent for optimization. Thus, before running the code, the code must be modified to create a function call to `optimize2d`, after passing in the appropriate functions. The function outputs an animation of the optimization process, and prints out the optimum value.

Function f3

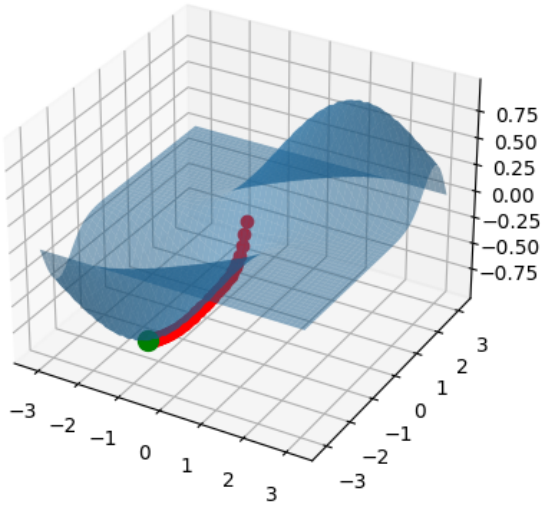
After passing in function `f3` to the generic function `optimize2d`, we get the final graph as shown below:



And the coordinates of the optimized point to be `(3.8914797970244126, 2.0000000001629625, 2.0001386891190123)`. The learning rate used here was *0.1* along x and *0.1* along y. The number of iterations were *100* and the initial point taken was *(x, y) = (3, 3)*.

Function f4

After passing in function `f4` to the generic function `optimize2d`, we get the final graph as shown below:



And the coordinates of the optimized point to be `(-1.5437464675140542, -1.5496773856949304, -0.9997418361239456)`. The learning rate used here was *0.1* along x and *0.1* along y. The number of iterations were *100* and the initial point taken was *(x, y) = (0, 0)*.