Assignment 5: Gradient Descent

Approach

- I have defined two functions, `oneDim(f,d_dx,lim)` for a single variable function and `twoDim(f, d_dx, d_dy, lim_x,lim_y,bestx=0,besty=0,lr=0.4)` for a two-variable function.
- Also `onestepderiv(frame)` is defined in each of these functions and is used to shift the value of parameter(s) by $\frac{df}{dx} \times lr$
- The program changes x to $\frac{df}{dx} \times lr$ and y to $\frac{df}{dy} \times lr$ on each iteration of `onestepderiv` driven by the `matplotlib.animation.FuncAnimation` function.
- The program also displays the values of parameters and the value of the function at that point every 3 iterations and every 5 iterations for one Dimensional and two Dimensional gradient descent respectively.
- I setup a learning rate for a function that determines how much the independent variables change per iteration. In every iteration of `onestepderiv(frame)`, the x value changes by the amount mentioned above.

Functions

1. `oneDim(f,d_dx,lim,filename="Animation.png")`

Parameters:

- f: The function to optimise
- d_dx: The derivative of the function
- lim: Bounds for the independent value
- filename: File to save animation to

Returns:

- Returns None
- Saves the Animation as a png file and displays it

2. `twoDim(f, d_dx, d_dy, lim_x,lim_y,bestx=0,besty=0,lr=0.4,filename="Animation.png")`

Parameters:

- f: The function to optimise
- d_dx: The Partial Derivative wrt x
- d_dy: The Partial Derivative wrt y
- lim_x: Bounds for x
- lim_y: Bounds for y
- bestx: Initial guess for x
- besty: Initial guess for y

- lr: Learning Rate
- filename: File to save animation to

Returns:

- Returns None
- Saves the Animation as a png file and displays it

Usage

The function calls have been commented in the last lines of the code. Uncomment and comment the ones required and run the program with the following command in the terminal:

`python EE22B086_Assignment5.py` or `python3 EE22B086_Assignment5.py`

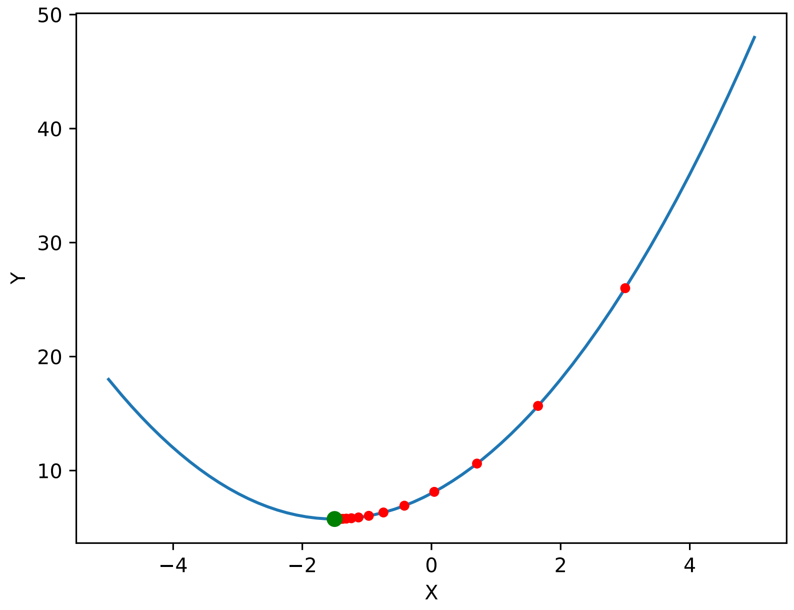
Question 1: $x^2 + 3x + 3$

The optimum value came out to be at $x = -1.499999999999$, which is very close to the actual value of $x = -1.5$

The optimum value of the function was $y = 5.75$

- The initial guess was $x = 3$. This was pretty random
- The bounds were $[-5, 5]$
- Learning Rate was 0.15

The animation is given below:



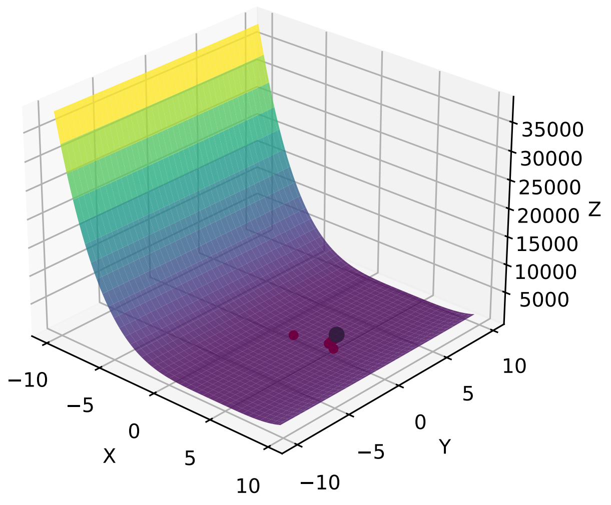
Question 2: $x^4 - 16x^3 + 96x^2 - 256x + y^2 - 4y + 262$

The optimum value came out to be at $(x, y) = (4.1006749700045, 1.9999999999)$

The minimum value of the function was $z = 2.0001027273381$

- Initial guess was $(x, y) = (2, 0)$. This was chosen to ensure that the function reaches the minimum. If x is lesser than this, the derivative is very high and hence, the function value does not decrease in an iteration.
- The bounds for x and y are $[-10, 10]$ and $[-10, 10]$
- Learning Rate was 0.1 for both x and y

The graph is given below:



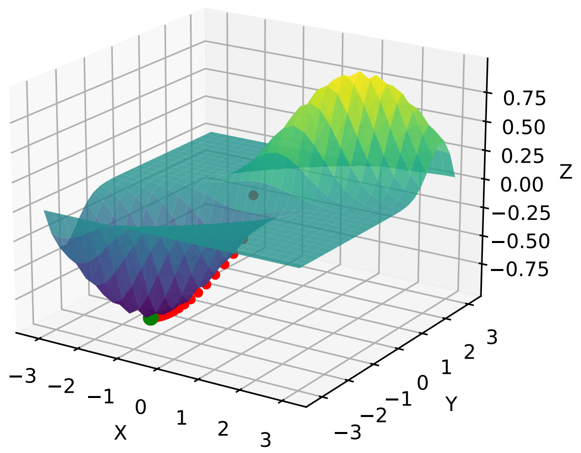
Question 3: $e^{-(x-y)^2} sin(y)$

The optimum was found at $(x, y) = (-1.570796306212, -1.570796310724)$

The minimum value of the function is $z = -0.9999999999999999$

- Initial guess was $(x, y) = (0, 0)$. This is one of the initial guesses that goes to the actual minimum of the function. For the values of x and y such that the function value ends up in a place perpendicular to the z axis, the values of x and y do not change and the minimum cannot be found.
- The bounds for x and y were $[-\pi, \pi]$ and $[-\pi, \pi]$

The graph is given below:



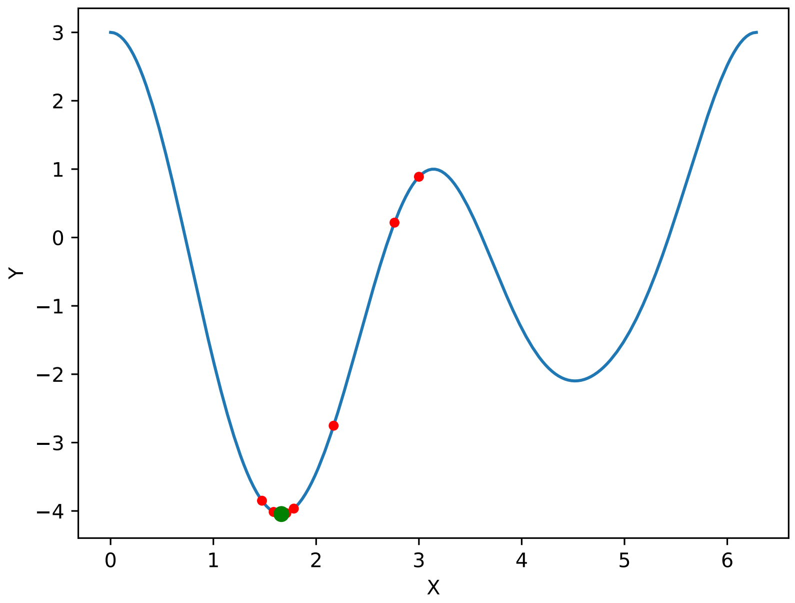
Question 4: $cos^4(x) - sin^3(x) - 4sin^2(x) + cos(x) + 1$

The optimum value came out to be at $x = 1.66166081204378$

The value of the function was $y = -4.045412051572552$

- Initial guess was $x = 3$. This was chosen to ensure that the minimum at the global minimum is reached to by the function and that the result doesn't go to the local minimum (greater than the global minimum)
- The bounds were $[0, 2\pi]$
- Learning Rate was 0.15

The animation is given below:



3
2
1
0
−1
−2
−3
−4
Y
0
1
2
3
4
5
6
X