**Assignment - Gradient Descent Algorithm**

***Student*** : Htet Aung Lynn

***Submitted Date*** : March 27th, 2024

***Due Date*** : April 5th, 2024

**Problem statement**

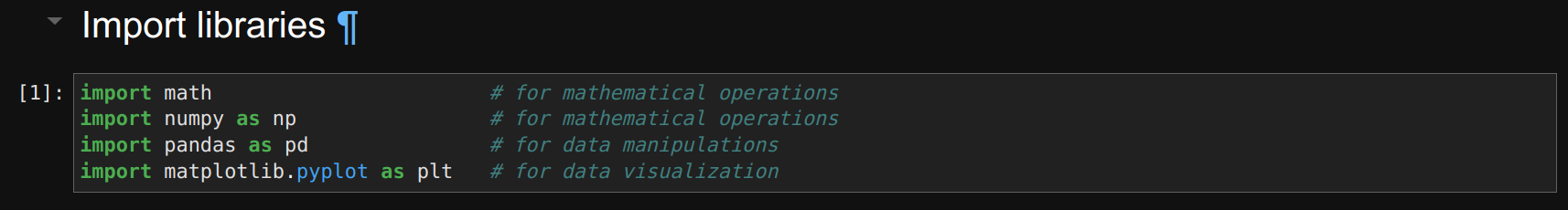
Suppose that you are asked to find the minimum point of the objective function within −4 ≤ 𝑥 ≤3:

**𝑓 (𝑥) = 𝑥4 + 𝑥3 − 6𝑥2 + 4𝑥 + 12**

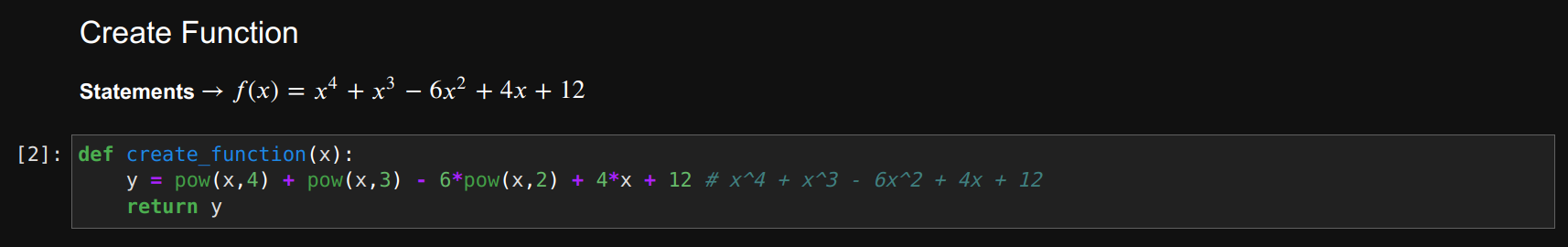
**Delivered Problems**

1. Create a plot showing the objective function within −4 ≤ 𝑥 ≤3.

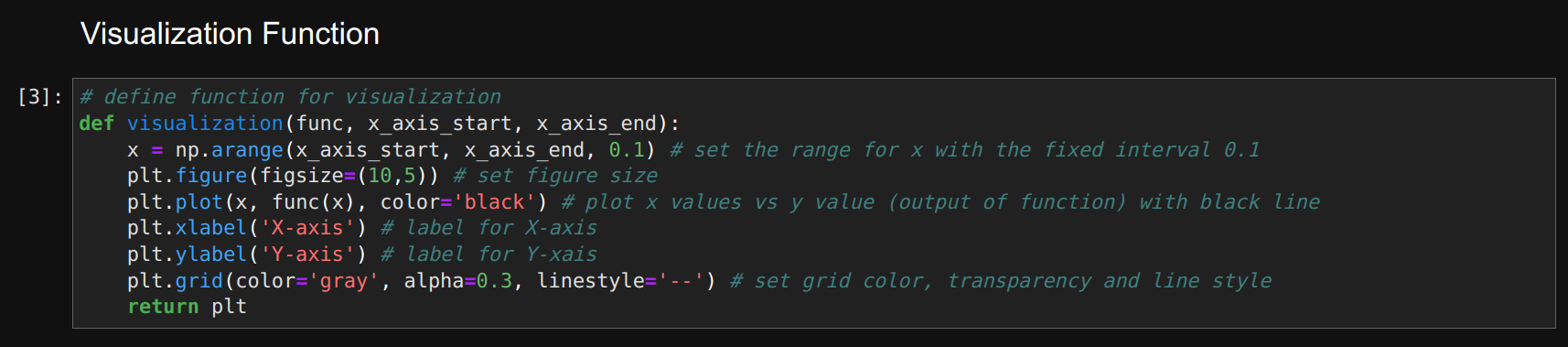
* Initialized by importing necessary libraries for the assignment:
  1. math
  2. numpy
  3. pandas and
  4. matplotlib



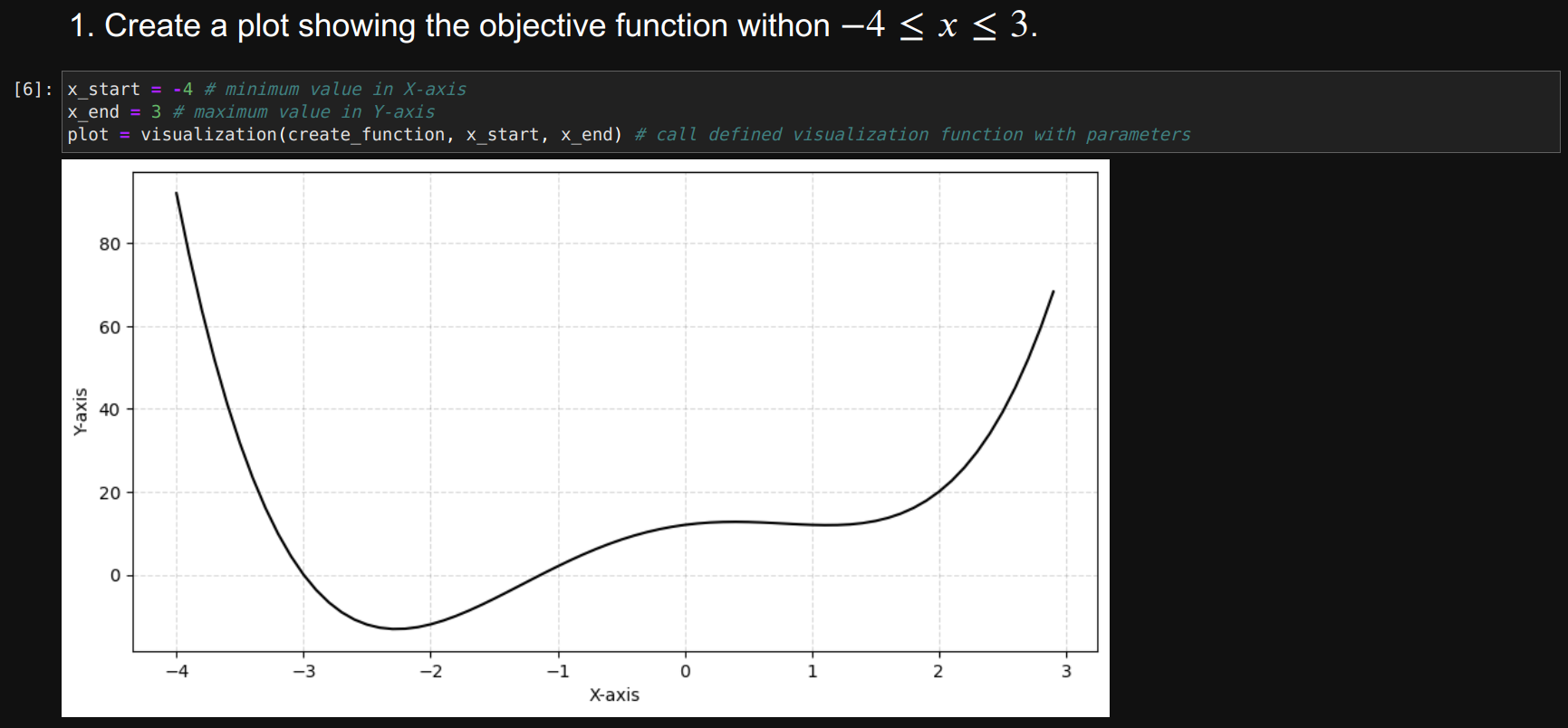
* Create a function for the given statement



* Create a function again for the purpose of visualization



* The output of the following codes are as per following:



2. Explain how you design your own gradient descent algorithm from scratch.

* My gradient descent algorithm is working as per following steps;

***Step 1***: assigned x\_initial value into variable x, as current x value

***Step 2***: plot given function with black colored line

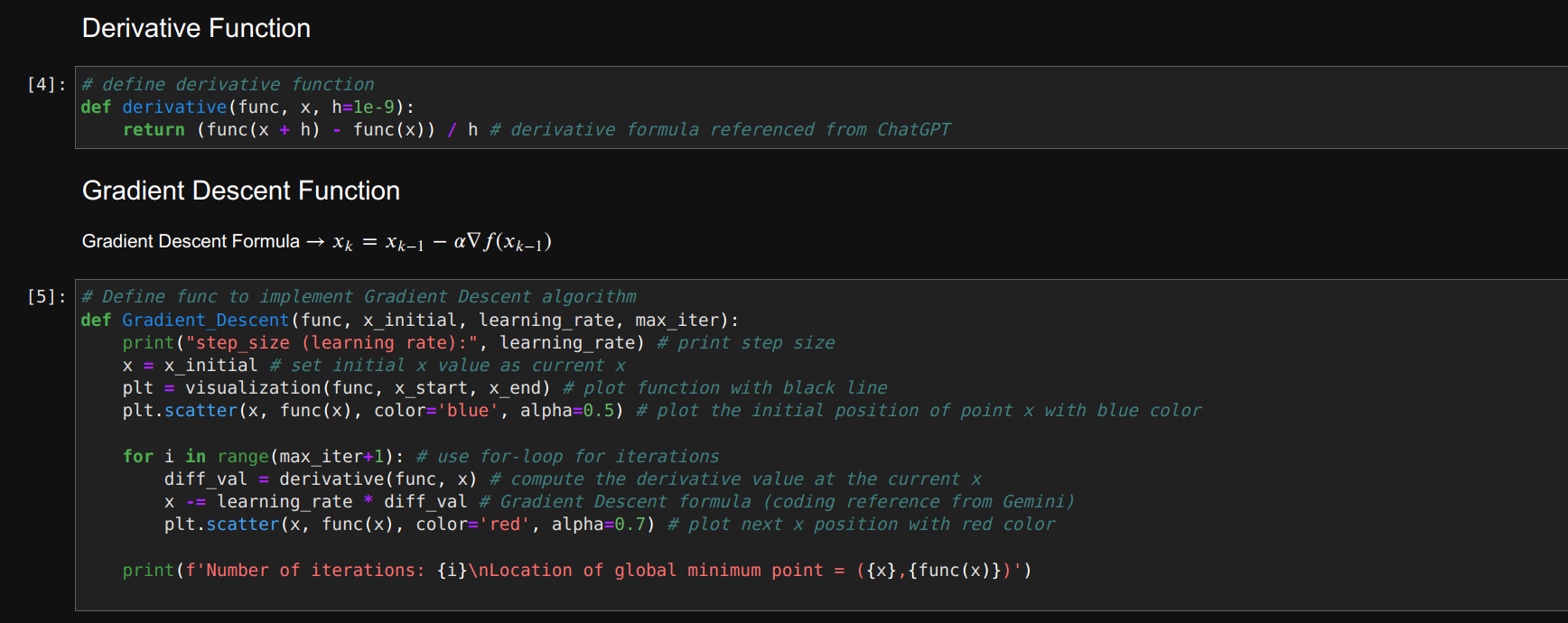
***Step 3***: plot initial position of point x with blue colored dot using scatter plot

***Step 4***: use for-loop for iteration

***Step 5***: call derivative function previously defined

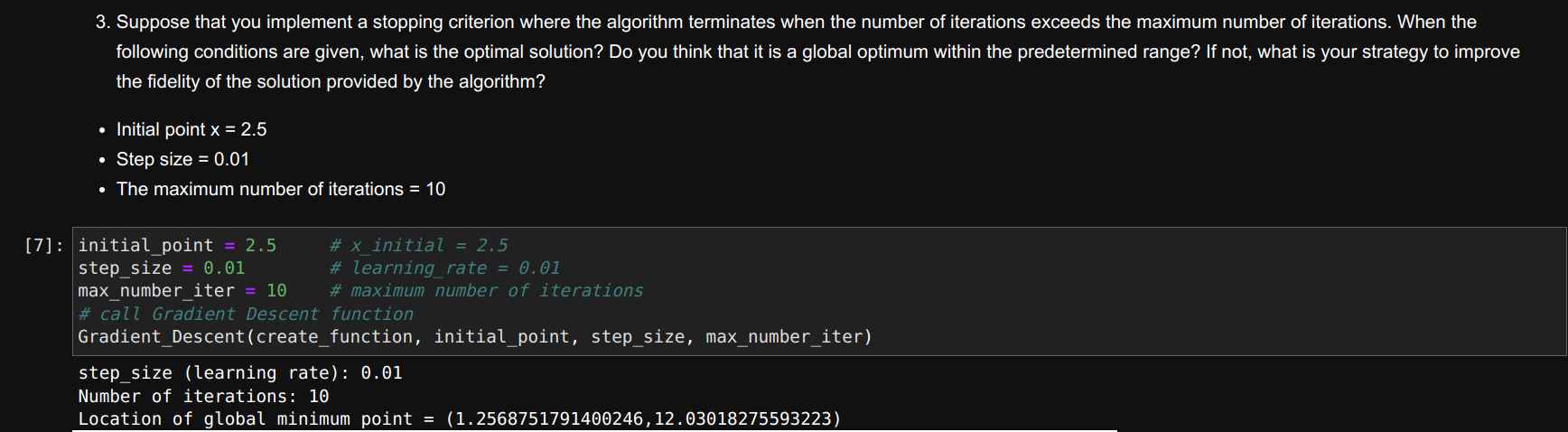
***Step 6***: get the new x value using Gradient Descent formula (coding reference from Gemini)

***Step 7***: plot the position of new x value with red colored dot using scatter plot

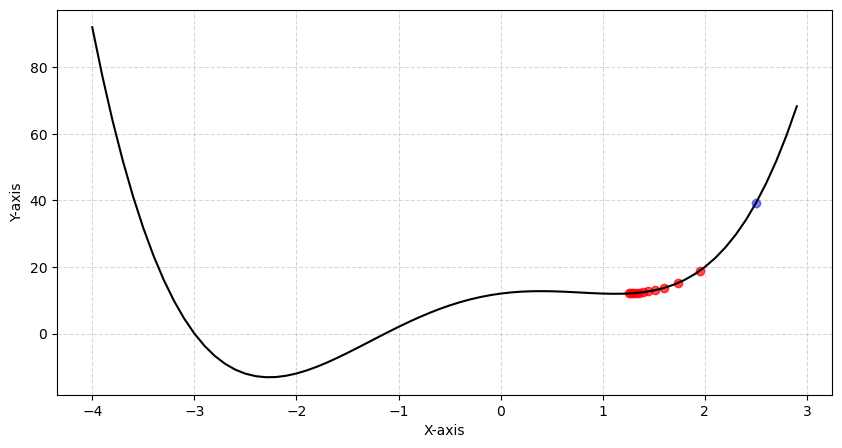


3. Suppose that you implement a stopping criterion where the algorithm terminates when the number of iterations exceeds the maximum number of iterations. When the following conditions are given, what is the optimal solution? Do you think that it is a global optimum within the predetermined range? If not, what is your strategy to improve the fidelity of the solution provided by the algorithm?

* Initial point x = 2.5
* Step size = 0.01
* The maximum number of iterations = 10
* The given parameters cannot outcome optimal solution since it cannot be found global minimum value on the function.



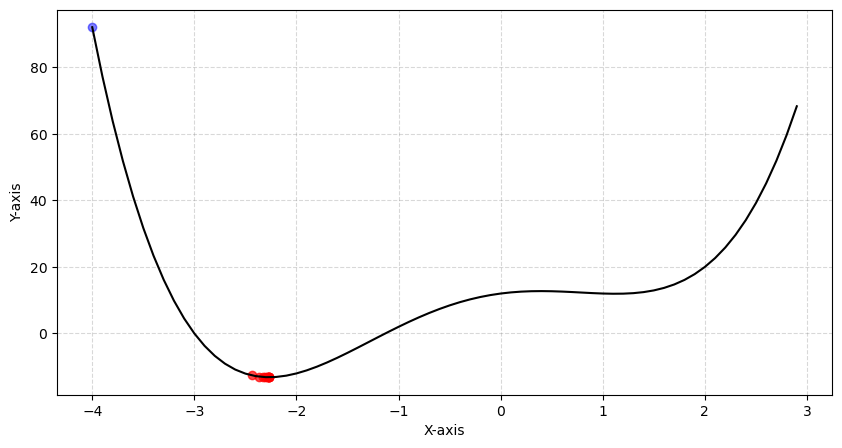
Output



* Strategy-1

The output for the following parameter setting can be resulted optimal outcomes.

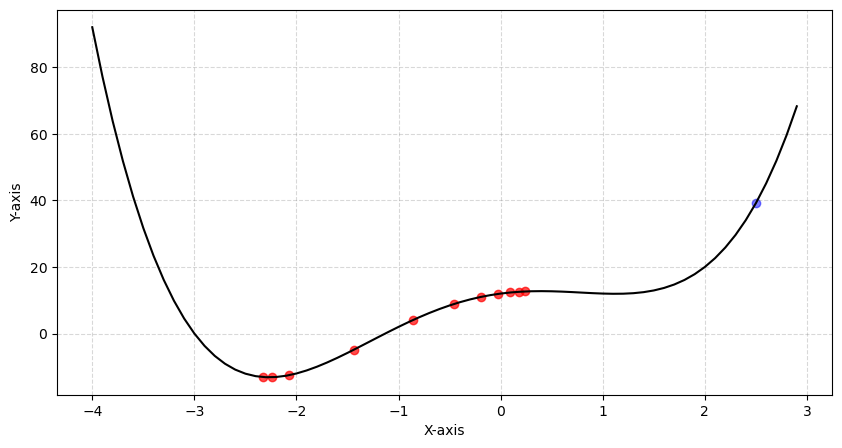
* + Initial point x = -4 (**improvement point**)
  + Step size = 0.01
  + The maximum number of iterations = 10



* Strategy-2

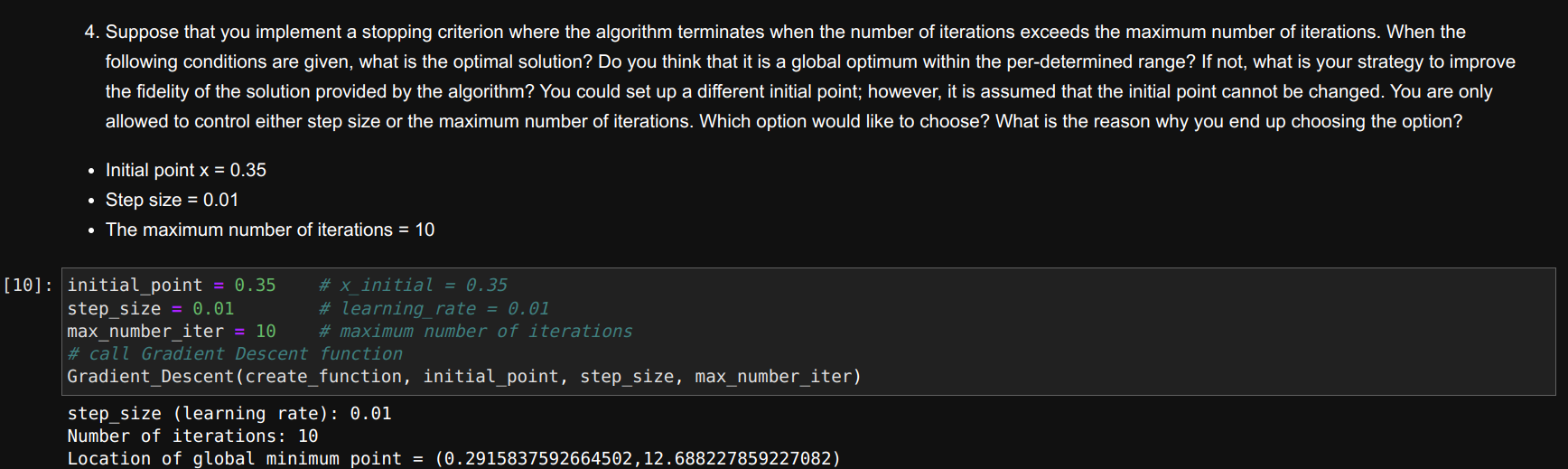
The output for the following parameter setting can be resulted optimal outcomes.

* + Initial point x = 2.5
  + Step size = 0.041 (**improvement point**)
  + The maximum number of iterations = 10

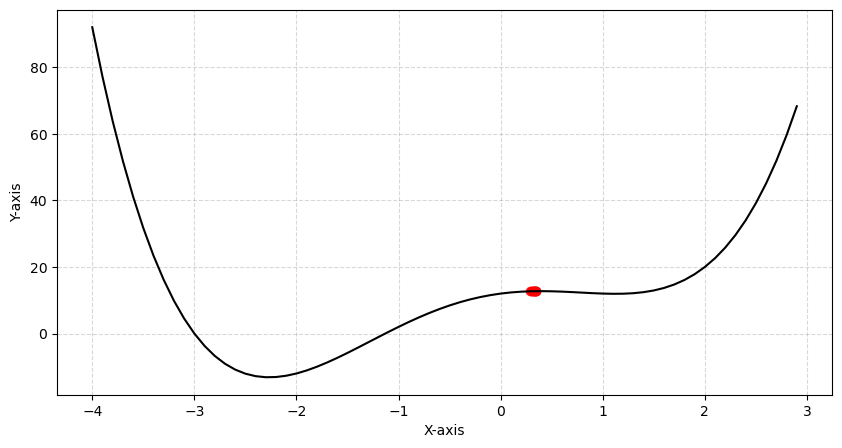


4. Suppose that you implement a stopping criterion where the algorithm terminates when the number of iterations exceeds the maximum number of iterations. When the following conditions are given, what is the optimal solution? Do you think that it is a global optimum within the per-determined range? If not, what is your strategy to improve the fidelity of the solution provided by the algorithm? You could set up a different initial point; however, it is assumed that the initial point cannot be changed. You are only allowed to control either step size or the maximum number of iterations. Which option would like to choose? What is the reason why you end up choosing the option?

* Initial point x = 0.35
* Step size = 0.01
* The maximum number of iterations = 10
* The given parameters cannot outcome optimal solution since it cannot be found global minimum value on the function.



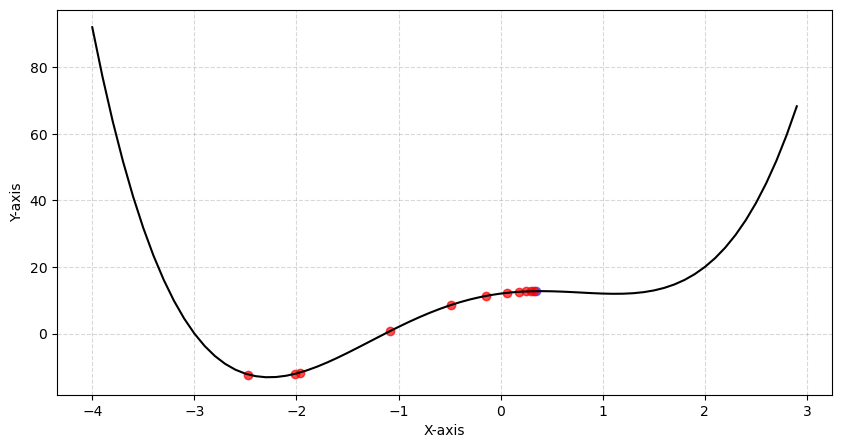
Output



* Option-1 (control step size)

The output for the following parameter setting can be resulted optimal outcomes.

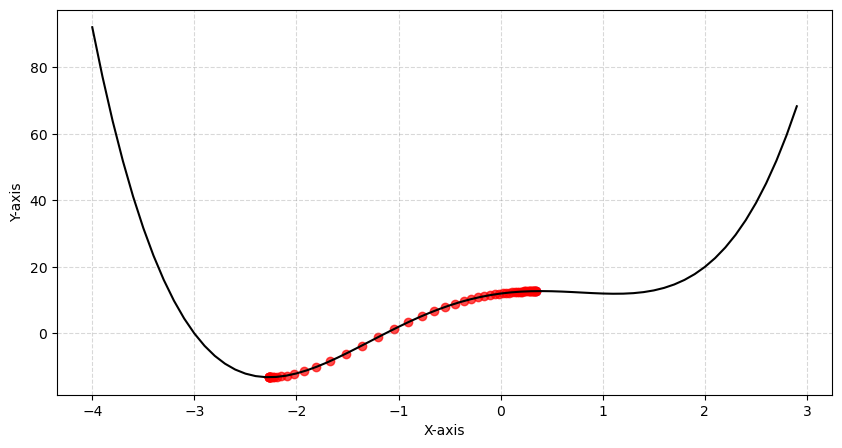
* + Initial point x = 0.35
  + **Step size = 0.06**
  + The maximum number of iterations = 10
  + Time taken = 0.024 seconds



* Option-2 (control maximum number of iterations)

The output for the following parameter setting can be resulted optimal outcomes.

* + Initial point x = 0.35
  + Step size = 0.01
  + **The maximum number of iterations = 60**
  + Time taken = 0.073 seconds



* From my point of view, adjusting step size (or) learning rate is better to control because it can reach the focus point more quickly while changing maximum number of iterations takes more time and high memory usage in calculation.

**References**

* ChatGPT
* Gemini