Nelder-Mead method implementation

Rafael Galiev

ČVUT–FIT

galieraf@fit.cvut.cz

May 23, 2023

1 Introduction

This report discusses an implementation of the Nelder-Mead method[3][4], a numerical method used to find the minimum or maximum of an objective function in a multidimensional space. The implementation allows for user-defined functions of one, two, or three variables.

2 Input data

The input data for this implementation comes directly from the user. Users can input their own functions of one, two, or three variables to evaluate. Variables, functions, and standard operations are all accepted input formats.

3 Used models

The primary model used in this implementation is the Nelder-Mead method itself. The method works by taking an initial 'guess' at a solution, and iteratively refining it by moving around the 'simplex' or shape defined by the points in the search space. The model was implemented using Python and utilized libraries such as numpy, matplotlib[6], and sympy[5].

4 Results

The implementation successfully finds the minimum or maximum of a user-defined function in a multidimensional space. The results obtained were as expected based on the mathematical properties of the functions tested. For example, as the Rosenbrock function[8], which is widely used to test optimization and minimization methods. The tests were implemented using the pytest library[2]. Comparison of the results with the expected ones was carried out using the standard deviation[7].

While the developed application and the implemented Nelder-Mead algorithm generally perform well for many functions, it's important to recognize that there are inherent limitations and challenges when dealing with certain types of mathematical functions, particularly those without defined extrema.

One such example we encountered was with the function:

$$f(x) = e^x \tag{1}$$

This function is a continuous, monotonically increasing function with no extrema. When applying the Nelder-Mead algorithm to this function, it returned a 'best' point of '-748', a result that can be difficult to interpret.

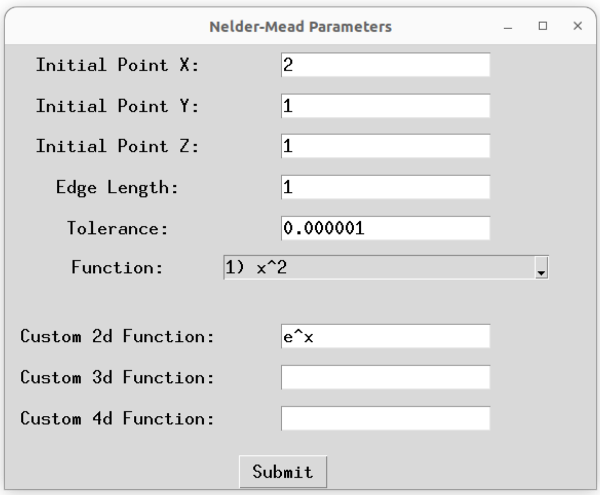

Figure 1: Entering

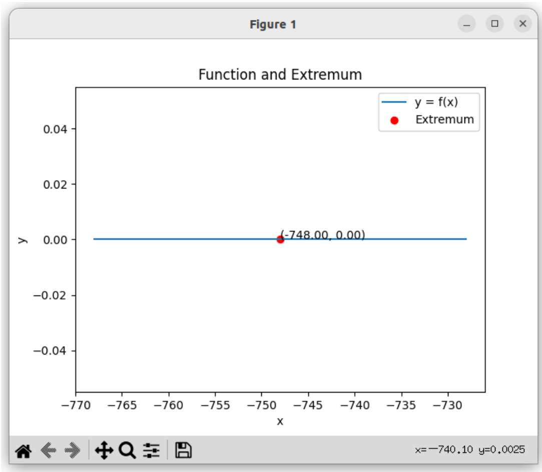

Figure 2: Result

The specific result of '-748' might be related to the limits of numerical stability and precision in computer computations. It is possible that at some point, the algorithm reached the limits of numerical precision for this function and/or the algorithm's parameters, which led to this result.

Therefore, it should be noted that the application is highly specific and depends on the user's data. It is best suited for those who have some understanding of mathematics and can discern whether a result is incorrect or unexpected. But of course I tried to write a good README[1] file that explains the program and important aspects and is understandable to everyone.

5 Conclusion

The implementation is a useful tool for performing optimization tasks on user-defined functions. Future improvements could include extending the implementation to handle more than three variables, improving the user interface, and optimizing the performance of the implementation.

I am very pleased with my project and am thrilled to have achieved reasonable results. I have a fondness for mathematics, and through Python, it's incredible to see the wonderful things one can create.

6 References

1. Guo, David. *Make a README* [Online]. 2019. Available from: <link>. [Accessed: May 9, 19:00 UTC, 2023].

2. Slapak, Martin. "render_tree.py" [Python script]. March 19, 2023. [Online]. Available to FIT CTU authorized students from: <link>. [Accessed: May 6, 03:02 UTC, 2023].

3. Wikipedia contributors. "Nelder-Mead method". In: *Wikipedia: The Free Encyclopedia.* 21 April 2023, 21:11 UTC. Available from: <link>. [Accessed: May 10, 16:00 UTC, 2023].

4. Himmelblau, D.M. (1972). "Applied nonlinear programming". The University of Texas, Austin, Texas: McGraw-Hill Book Company.

5. SymPy Development Team. SymPy Documentation v1.12. [Online]. 2023. Available from: <link> [Accessed: May 10, 07:00 UTC, 2023]

6. Matplotlib Development Team. Matplotlib Documentation. [Online]. 2023. Available from: <link>. [Accessed: May 10, 10:00 UTC, 2023].

7. Wikipedia contributors. "Standard deviation". In: *Wikipedia: The Free Encyclopedia.* Available from: <link>. [Accessed: May 7, 14:37 UTC, 2023].

8. Wikipedia contributors. "Rosenbrock function". In: *Wikipedia: The Free Encyclopedia.* Available from: <link>. [Accessed: May 7, 16:25 UTC, 2023].