FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION
OF HIGHER EDUCATION
ITMO UNIVERSITY

Report
on the practical task No. 3
"Algorithms for unconstrained nonlinear optimization. First- and second-order methods"

Performed by
Chernobrovkin Timofey (412642)
Academic group J4133c
Accepted by
Dr Petr Chunaev

St. Petersburg
2023

Goal

The use of first- and second-order methods (Gradient Descent, Non-linear Conjugate Gradient Descent, Newton's method and Levenberg-Marquardt algorithm) in the tasks of unconstrained nonlinear optimization.

Problem

Generate random numbers $\alpha \in (0,1)$ and $\beta \in (0,1)$. Furthermore, generate the noisy data $\{x_k, y_k\}$, where $k = 0, \ldots, 100$, according to the following rule:

$$y_k = \alpha x_k + \beta + \delta_k, x_k = \frac{k}{100},$$

where $\delta_k \sim N(0,1)$ are values of a random variable with standard normal distribution. Approximate the data by the following linear and rational functions:

1. $F(x, a, b) = ax + b$ (linear approximant),
2. $F(x, a, b) = \frac{a}{1+bx}$ (rational approximant),

by means of least squares through the numerical minimization (with precision $\varepsilon = 0.001$) of the following function:

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2.$$

To solve the minimization problem, use the methods of Gradient Descent, Conjugate Gradient Descent, Newton's method and Levenberg-Marquardt algorithm. If necessary, set the initial approximations and other parameters of the methods. Visualize the data and the approximants obtained in a plot separately for each type of approximant so that one can compare the results for the numerical methods used. Analyze the results obtained (in terms of number of iterations, precision, number of function evaluations, etc.) and compare them with those from Task 2 for the same dataset)

Theory

Optimization methods are numerical methods for finding optimal (in a sense) results of objective functions, for example, within the framework of mathematical models of certain processes. Optimization methods are widely used in data analysis and machine learning.

Let the objective function $f = f(x)$ be given, where x is, generally speaking, a multidimensional vector from some subset Q of the Euclidean space $\mathbb{R}^m$. The subset Q can be either limited or, in particular, coincide with the entire space $\mathbb{R}^m$. Below, for brevity, we will consider only the problem of minimizing the function f on the set Q (we can move from minimization to maximization by considering $F(x) = -f(x)$ instead of $f(x)$).

By solving the optimization (minimization) problem $f(x) \rightarrow min_{x \in D}$ we mean finding $x^* \in Q$ such that $f(x^*) = min_{x \in Q} f(x)$. There is a special notation for $x^*: x^* = arg\ min_{x \in Q} f(x)$. If x^* is found, then, naturally, $f(x^*)$ can also be found.

The stated formulation of the optimization problem implies the search for a global minimum of f on Q. However, below we will also consider the search for a local minimum of f on Q, when the value of $f(x^*)$ is minimal only in a certain neighborhood of the point x^*. Note that often the search for a local minimum is much simpler than a global one.

This lab work uses first-order methods - gradient descent, conjugate gradient method, and second-order methods - Newton's method and Levenberg-Marquardt algorithm.

The optimization problem associated with linear approximation has a single solution, and therefore we should expect that the above methods will yield very similar optimal values for a and b, regardless of the choice of initial approximations. In the case of rational approximation, however, significant nonlinearities arise, and even the choice of initial approximation can significantly affect the result.

Gradient descent is a numerical method of finding the local minimum or maximum of a function by moving along a gradient, one of the main numerical methods of modern optimization. It is actively used in computational mathematics not only for direct solution of optimization (minimization) problems, but also for problems that can be rewritten in the optimization language. The gradient descent method can be used for optimization problems in infinite-dimensional spaces, for example, for numerical solution of optimal control problems. There has been particularly great interest in gradient methods in recent years due to the fact that gradient descent and its stochastic/randomized variants underlie almost all modern learning algorithms developed in data analysis.

The conjugate gradient method is an iterative method for unconditional optimization in multidimensional space. The main advantage of the method is that it solves a quadratic optimization problem in a finite number of steps. Therefore, we first describe the conjugate gradient method for quadratic functional optimization, derive the iterative formulas, and give estimates of the convergence rate. After that, we show how the conjugate method is generalized to optimize an arbitrary functional, consider various variants of the method, and discuss convergence.

Newton's method is an iterative numerical method of finding the root (zero) of a given function. The solution is found by constructing successive approximations and is based on the principles of simple iteration. The method has quadratic convergence. A modification of the method is the method of chords and tangents. Newton's method can also be used to solve optimization problems in which it is required to determine the zero of the first derivative or gradient in the case of multidimensional space.

Levenberg-Marquardt algorithm is an optimization method aimed at solving least squares problems. It is an alternative to Newton's method. It can be considered as a combination of the latter with the method of gradient descent or as a method of confidence regions.

Materials and methods

In this task, all calculations were performed on the student's personal laptop. The work was performed in the Python programming language.

Results

Based on the task at hand, several first- and second-order methods were implemented. First, the data were approximated by a linear function. The results obtained for the gradient descent methods are shown in Figure 1, for the conjugate gradient method in Figure 2, for Newton's method in Figure 3 and for the Levenberg-Marquardt algorithm in Figure 4. Also for the mentioned algorithms we have performed calculations with approximation by rational function, the results are shown in Figures 5, 6, 7, 8, respectively.

General data for all considered algorithms with linear function approximation are presented in Table 1, and for rational function in Table 2.

In addition, the assignment requires comparing the results obtained in this work with the results of the previous assignment. Graphs for comparison are given in Figures 9, 10, 11 for the linear function and in Figures 12, 13, 14 for the rational function, respectively. The overall data for all algorithms of the past assignment with linear function approximation are presented in Table 2, and for the rational function in Table 3.

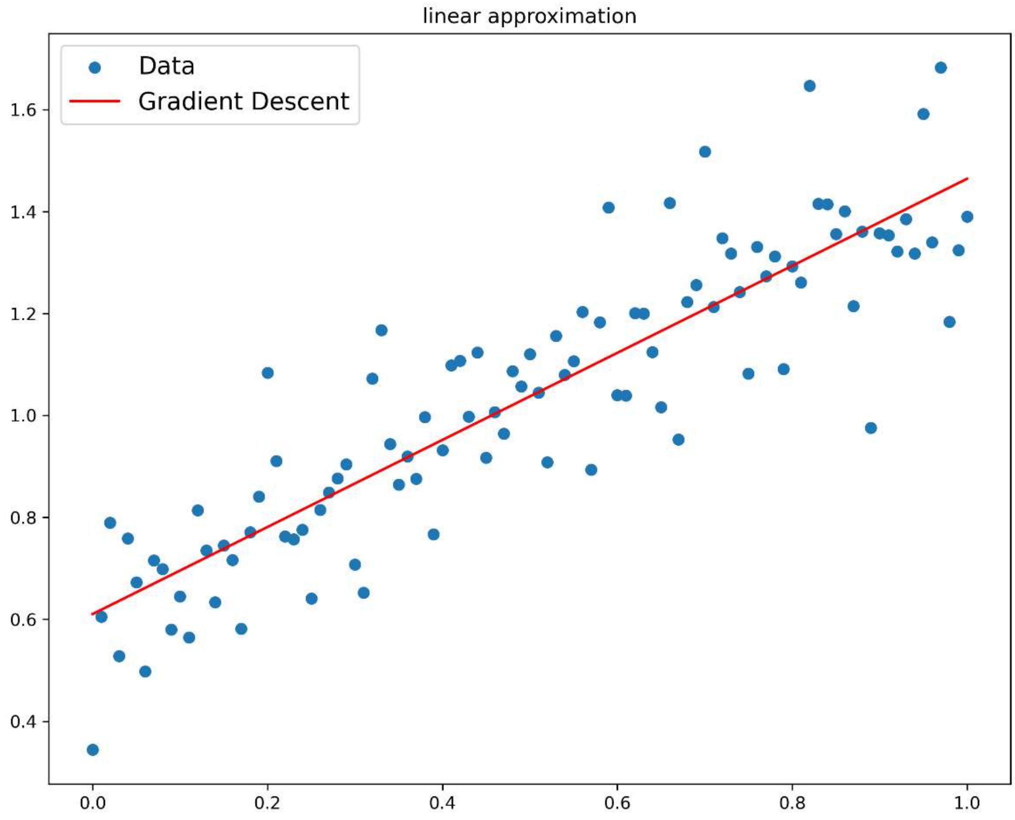

Figure 1 – Gradient descent linear approximation

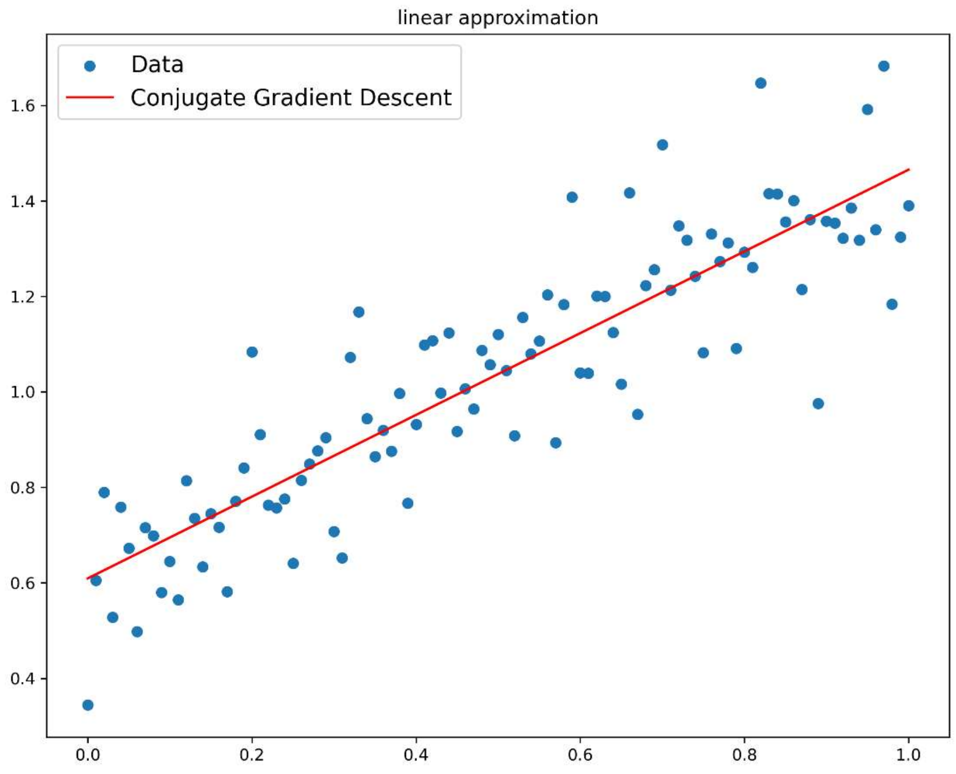

Figure 2 – Conjugate gradient descent linear approximation

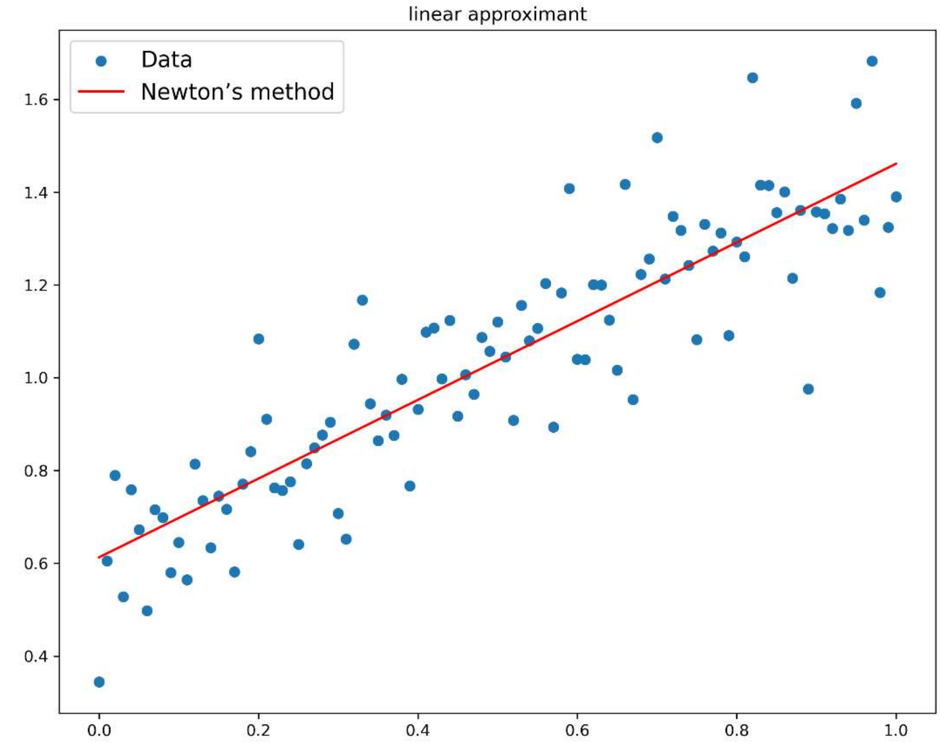

Figure 3 – Newton's method linear approximation

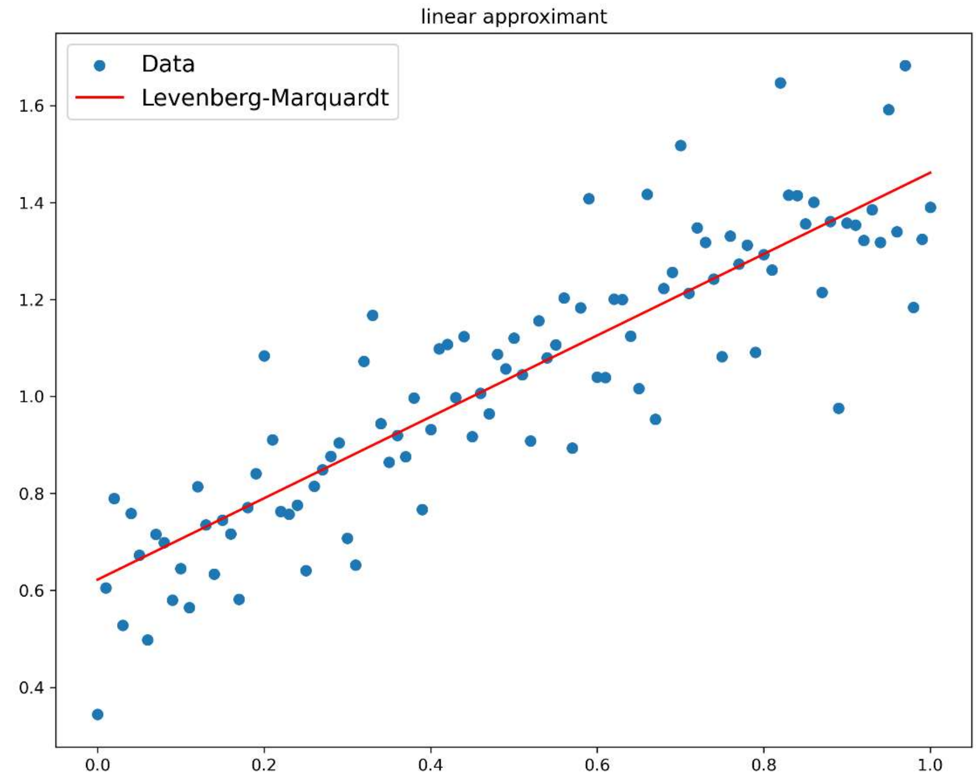

Figure 4 – Levenberg-Marquardt algorithm linear approximation

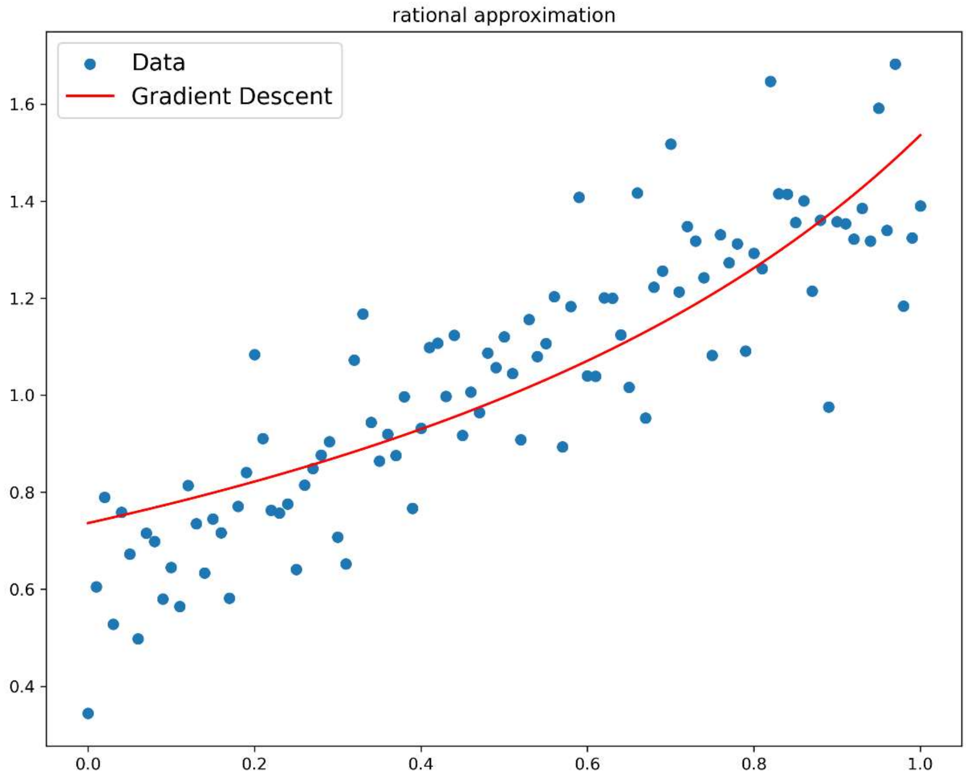

Figure 5 – Gradient descent rational approximation

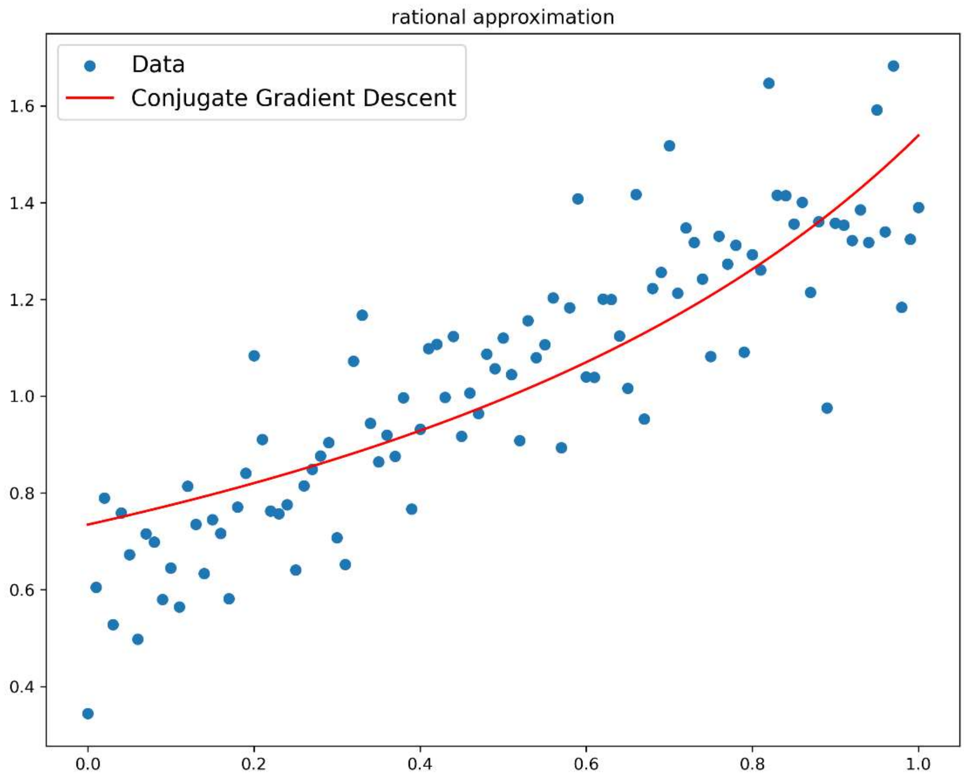

Figure 6 – Conjugate gradient descent rational approximation

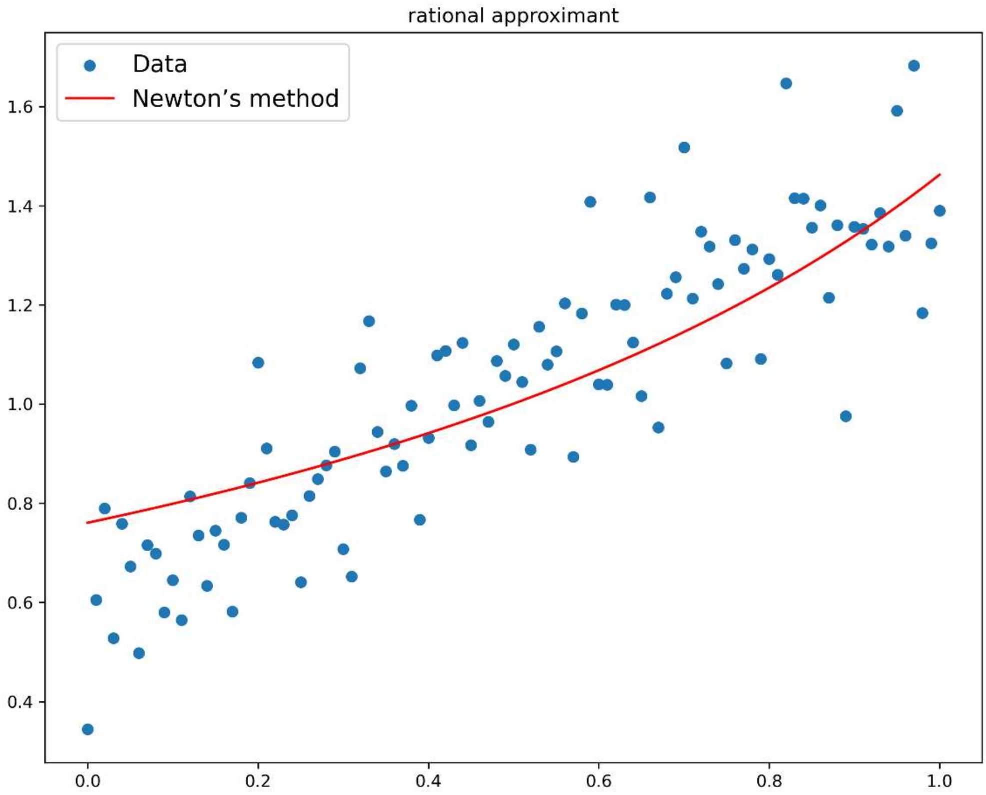

Figure 7 – Newton’s method rational approximation

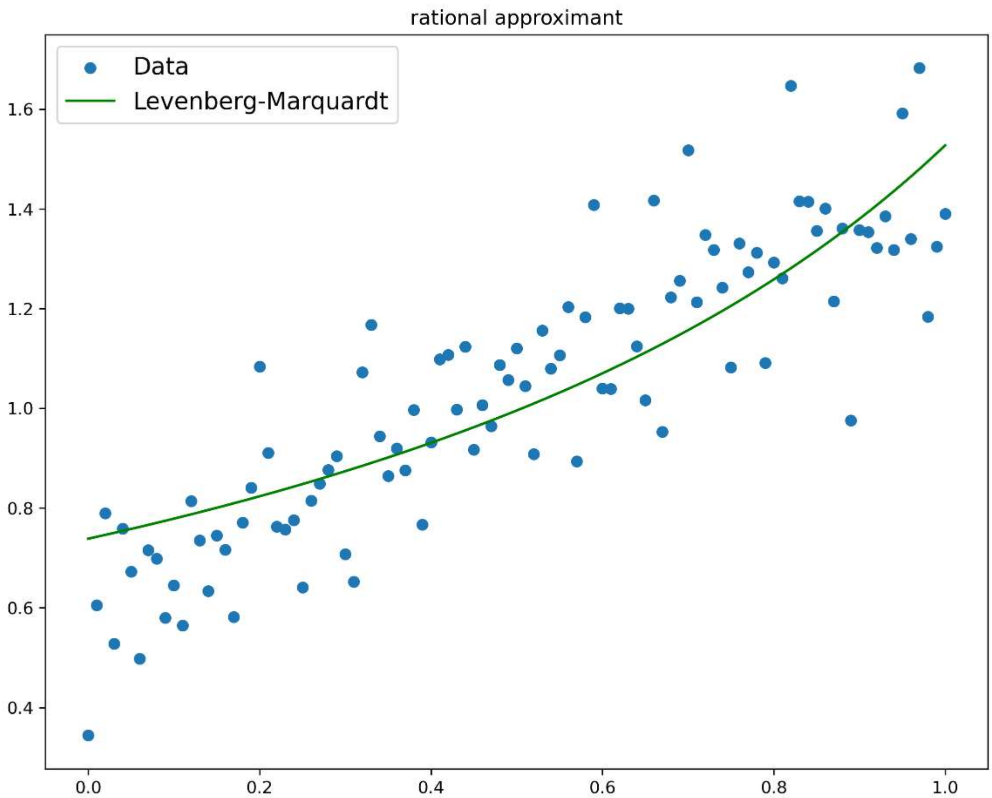

Figure 8 – Levenberg-Marquardt algorithm rational approximation

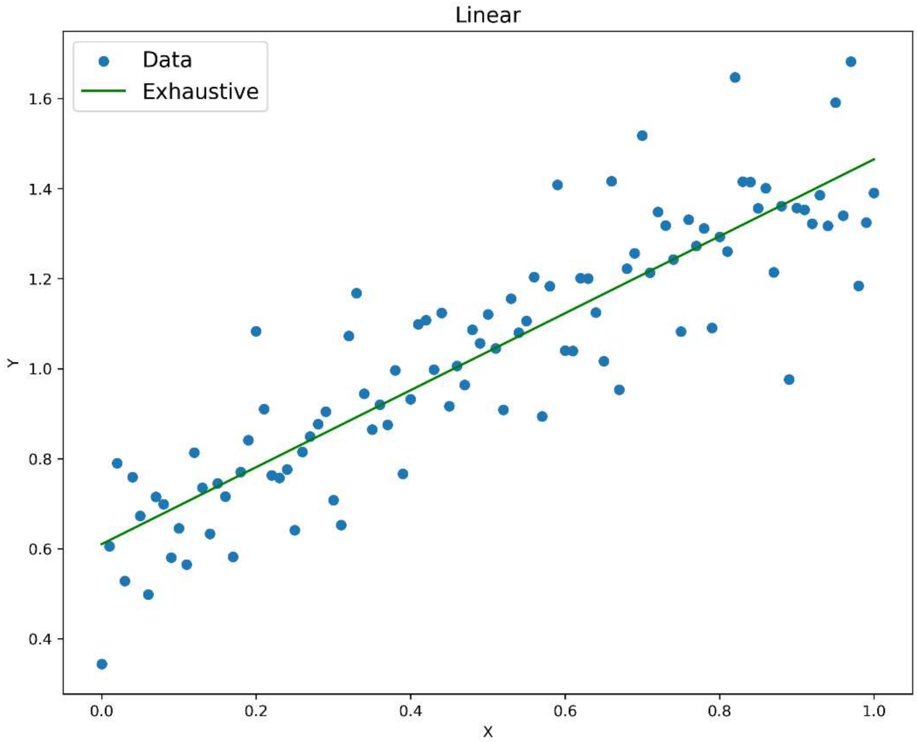

Figure 9 – Exhaustive search linear approximant

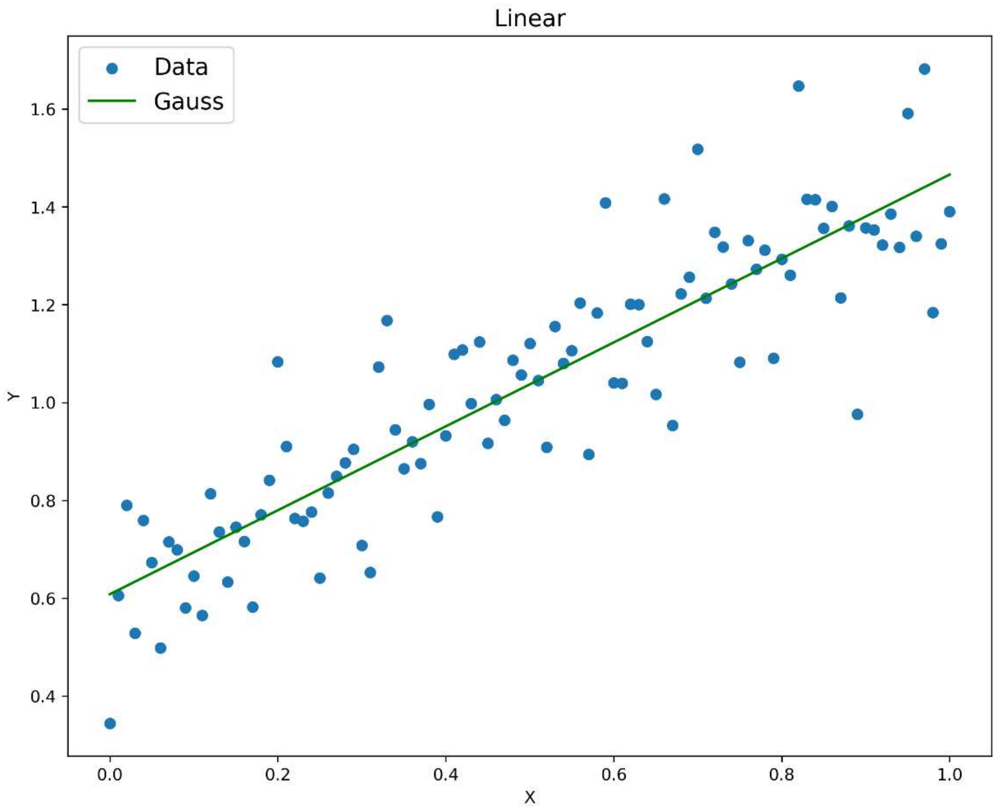

Figure 10 – Gauss linear approximant

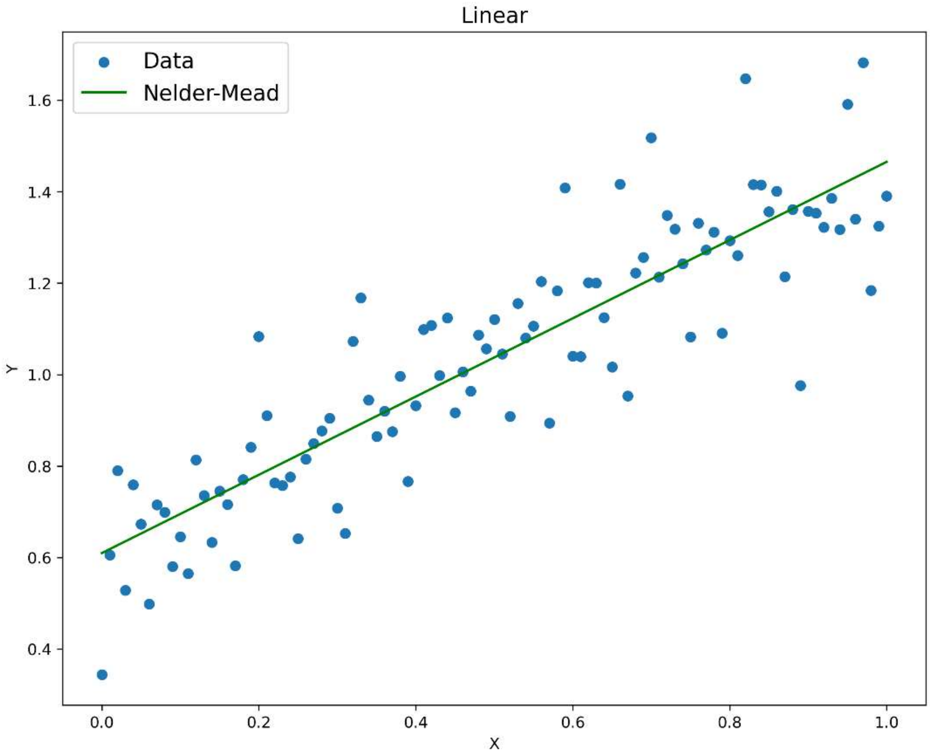

Figure 11 – Nelder-Mead linear approximant

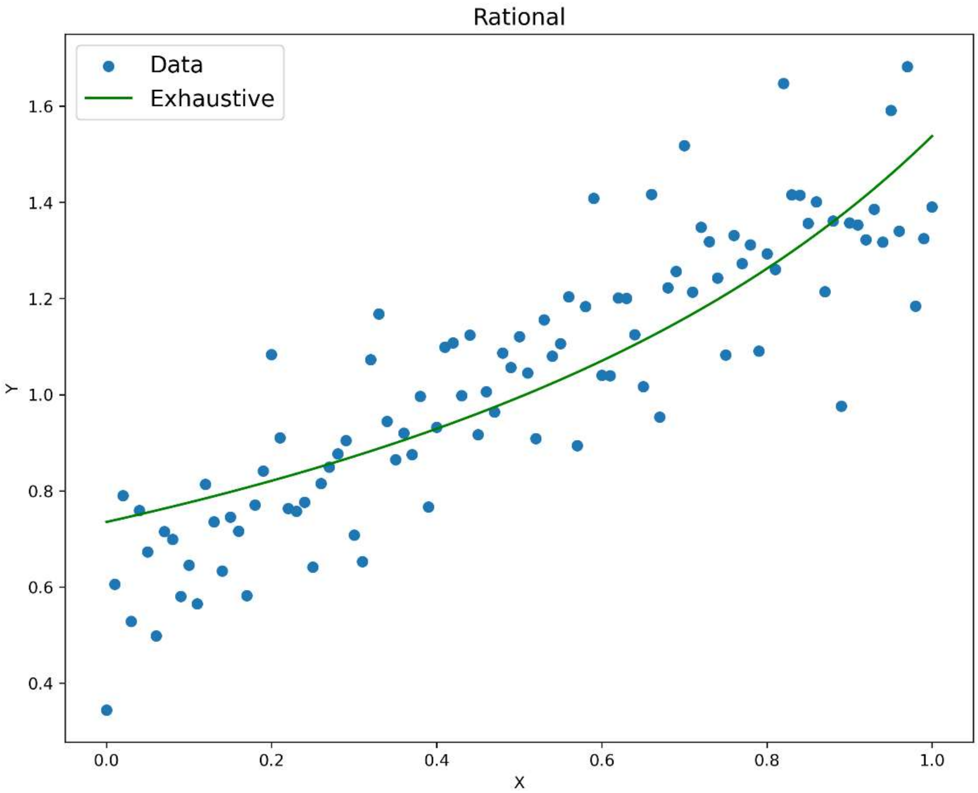

Figure 12 – Exhaustive search rational approximant

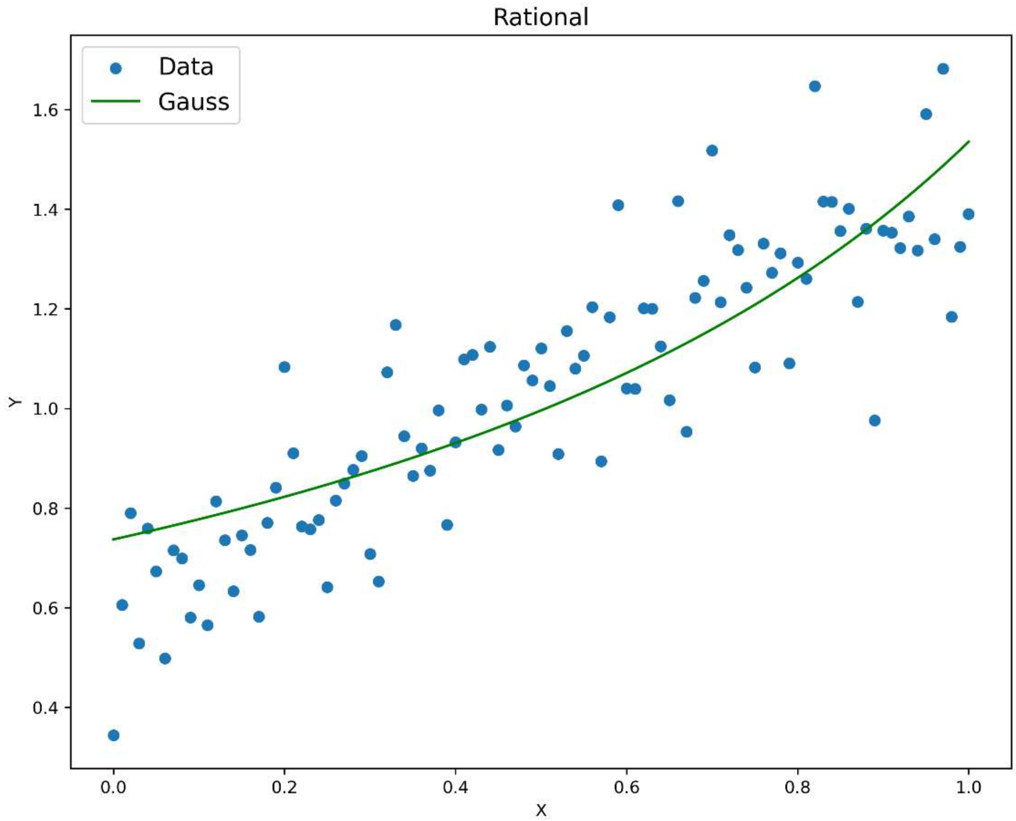

Figure 13 – Gauss rational approximant

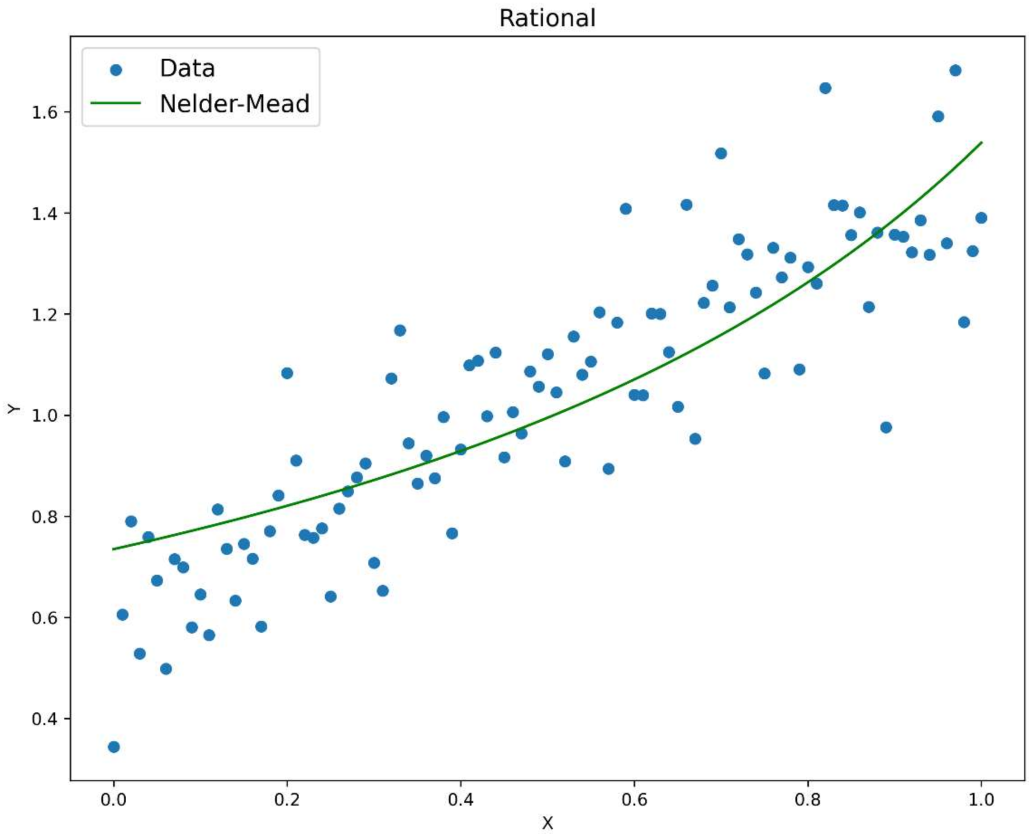

Figure 14 – Nelder-Mead rational approximant

Table 1 – Data of linear approximant

method	a	b	f-calculations	number of iterations
Gradient Descent	0.8540	0.6103	8002	4001
Conjugate Gradient Descent	0.8559	0.6093	15	2
Newton’s	0.8481	0.6129	13	11
Levenberg-Marquardt	0.8393	0.6221	31	3

Table 2 – Data of rational approximant

method	a	b	f-calculations	number of iterations
Gradient Descent	0.7363	-0.5205	2002	1001
Conjugate Gradient Descent	0.7346	-0.5225	88	12
Newton’s	0.7606	-0.4798	27	3
Levenberg-Marquardt	0.7385	-0.5164	40	3

Table 3 – Task 2 data of linear approximant

method	a	b	f-calculations	number of iterations
Exhaustive search	0.8550	0.6100	1000000	1000000
Gauss	0.8580	0.6080	30030	30
Nelder-Mead	0.8556	0.6096	39	21

Table 4 – Task 2 data of rational approximant

method	a	b	f-calculations	number of iterations
Exhaustive search	0.7350	-0.5220	1000000	1000000
Gauss	0.9990	-0.9990	31000	32
Nelder-Mead	0.7348	-0.5224	40	78

Conclusion

Thus, in this assignment I studied the application of first and second order methods (gradient descent, conjugate gradient method, Newton's method and Levenberg-Marquardt.

According to the obtained results, the least number of iterations with linear approximation has Conjugate Gradient Descent, and the least number of calculations of Newton's method function. With rational approximation the least number of iterations possesses Newton's method and Levenberg-Marquardt, and the least number of calculations of function Newton's method.

Сравнивая результаты с прошлым заданием можно сказать, что полученные графики функций получились практически одинаковые по линейной аппроксимации и с незначительными различиями по рациональной аппроксимации. Сравнивая количество итераций видим, что методы первого и второго порядка явно быстрее прямых методов. Однако сравнивая по количествам вычислений функции метод Nelder-Mead близок к методу Levenberg-Marquardt.

Appendi

GitHub link:
https://github.com/LesostepnoyGnom/Homework/blob/main/Task_3_20.09.23/Task_3_Chernobrovkin_J4133c.py