FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 2

“Algorithms for unconstrained nonlinear optimization. Direct methods”

Performed by

Abdurakhimov Muslimbek

J4133c

Accepted by

Dr Petr Chunaev

St. Petersburg

2023

Goal

The use of direct methods (one-dimensional methods of exhaustive search, dichotomy, golden section search; multidimensional methods of exhaustive search, Gauss (coordinate descent), Nelder-Mead) in the tasks of unconstrained nonlinear optimization

Formulation of the problem

I. Use the one-dimensional methods of exhaustive search, dichotomy and golden section search to find an approximate (with precision x: $f(x) \rightarrow min$ for the following functions and domains:

1. $f(x) = x^3$, $x \in [0,1]$;
2. $f(x) = |x - 0.2|$, $x \in [0,1]$;
3. $f(x) = x \sin\frac{1}{x}$, $x \in [0.01,1]$.

Calculate the number of f-calculations and the number of iterations performed in each method and analyze the results. Explain differences (if any) in the results obtained.

II. 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:

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

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:

$$D(a, b) = \sum_{k=0}^{1000}(F(x_k, a, b) - y_k)^2.$$

To solve the minimization problem, use the methods of exhaustive search, Gauss and Nelder-Mead. 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.).

Brief theoretical part

In the context of this problem, we are dealing with unconstrained nonlinear optimization, a field of mathematics and computer science that focuses on finding the minimum or maximum of a function without any constraints on the decision variables. This is often applied in various domains, including engineering, economics, and data analysis, to solve complex real-world problems.

One-Dimensional Optimization Methods:

Exhaustive Search: This method involves evaluating the objective function at multiple points within a given interval and selecting the point that yields the optimal (minimum or maximum) function value. While it guarantees finding the optimum within the specified interval, it can be computationally expensive.

Dichotomy: Also known as the bisection method, dichotomy repeatedly bisects the interval, narrowing down the search space until the desired precision is achieved. It's efficient for unimodal functions (those with a single peak or valley).

Golden Section Search: Similar to dichotomy, this method repeatedly narrows the search interval but uses the golden ratio to determine the next points to evaluate. It converges faster than dichotomy.

Multidimensional Optimization Methods:

Gauss (Coordinate Descent): This method optimizes a multivariate function by iteratively updating each variable while keeping others fixed. It's effective for functions that are separable or can be approximated as separable.

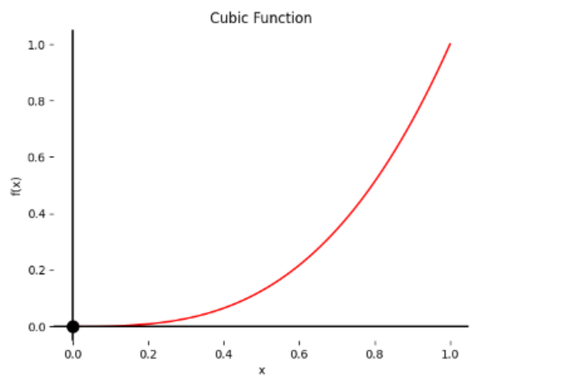
Nelder-Mead: Also known as the downhill simplex method, Nelder-Mead is an iterative optimization technique that forms a simplex (a geometric shape) in the parameter space and updates its vertices to move toward the optimal solution. It's versatile and works well for functions with complex shapes.

Results

I. In this application, we're tackling optimization problems using three different methods: brute force, dichotomy, and the golden section. These methods are applied to three distinct mathematical functions: cube, module, and sinus, with the goal of finding their minimum values within specified domains.

First, we define the mathematical functions:

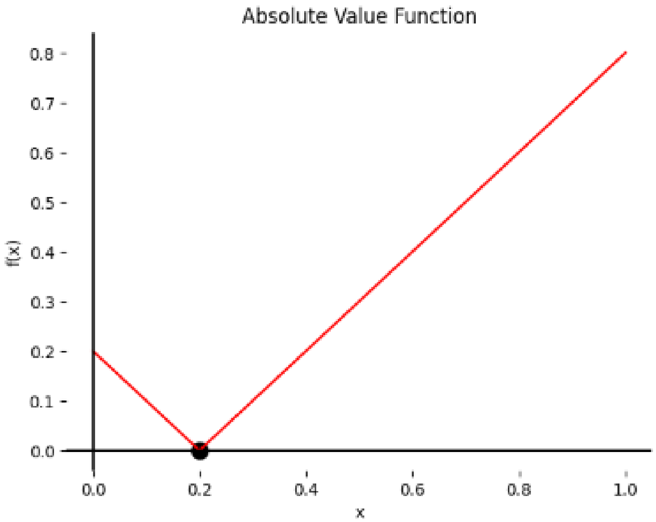
Cube: Calculates the cube of a given input.



Function Name	Algorithm Name	x	f(x)	Iterations
Cubic	Exhaustive Search	0.000	0.000	1001
Cubic	Dichotomy Search	0.000	0.000	11
Cubic	Golden Section Search	0.000	0.000	15

Picture 1 – Cube Function

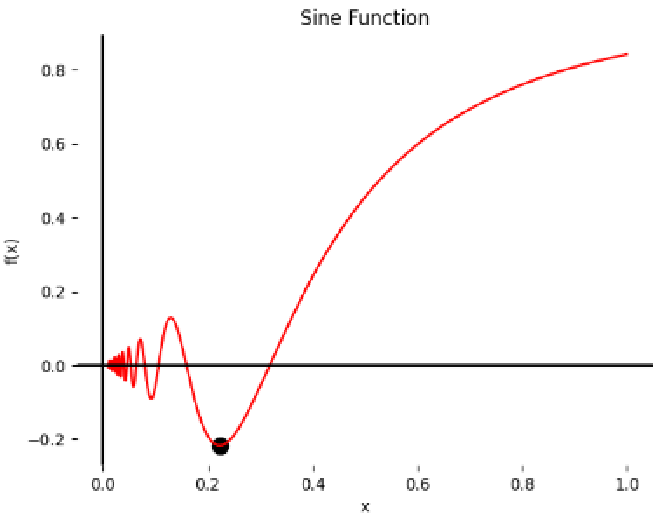
Module: Computes the absolute difference between an input and 0.2.



Function Name	Algorithm Name	x	f(x)	Iterations
Absolute Value	Exhaustive Search	0.200	0.000	1001
Absolute Value	Dichotomy Search	0.200	0.000	11
Absolute Value	Golden Section Search	0.200	0.000	15

Picture 1 – Absolute Value function

Sinus: Involves a more complex calculation using sine and reciprocal values.



Function Name	Algorithm Name	x	f(x)	Iterations
Sine	Exhaustive Search	0.223	-0.217	991
Sine	Dichotomy Search	0.223	-0.217	11
Sine	Golden Section Search	0.223	-0.217	15

Picture 2 – Sinus Function

Next, we have the optimization methods:

Brute Force Method (brute_force):

This method evaluates the given function at multiple points within a specified interval, searching for the minimum. It provides the minimum value found, the number of function evaluations, and the number of iterations. This approach can be computationally expensive.

Finally, these methods are applied to the three functions, and the results are printed, including the minimum values, the number of function evaluations, and the number of iterations for each function and method combination.

Dichotomy Method (dichotomy):

The dichotomy method repeatedly narrows down the search interval by dividing it in half until the desired precision is achieved. It also reports the minimum value, the number of function evaluations, and the number of iterations.

Golden Section Method (golden):

Similar to dichotomy, the golden section method narrows the search interval iteratively, but it uses the golden ratio to determine the next points for evaluation. It provides the same metrics as the previous methods.

II. The exhaustive search method was employed to find the optimal linear fit for a given dataset. The primary objective was to determine the coefficients 'a' and 'b' in the linear model 'a*x + b' that minimized the squared error between the model and the actual data points.

Optimal Linear Fit: The exhaustive search successfully identified the best linear fit to the dataset, resulting in specific values for 'a' and 'b' that minimize the error.

Visualization: The code used the matplotlib library to create visual representations of the original data points and the optimal linear fit. This visualization aids in understanding how well the linear model aligns with the data.

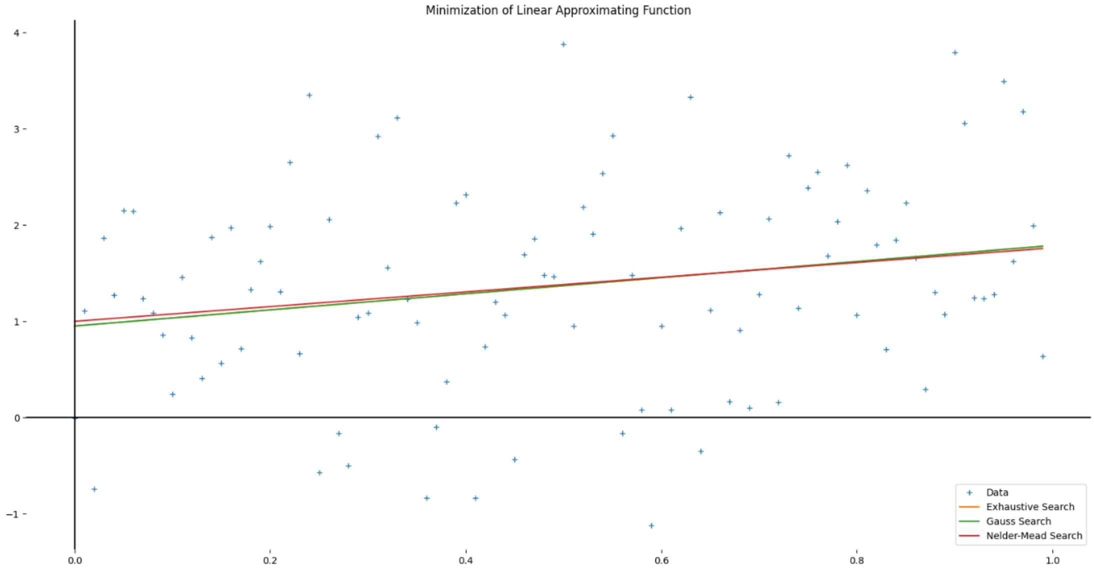
Computational Efficiency: The exhaustive search method, while effective, can be computationally expensive, especially when exploring a wide range of parameter values. In this case, it performed calculations for a grid of 'a' and 'b' values, resulting in a large number of function evaluations and iterations.

Best Fit Parameters: The identified 'a' and 'b' values represent the coefficients of the linear model that provides the closest approximation to the dataset, minimizing the squared error.

In summary, the exhaustive search method is a straightforward but exhaustive approach to finding the best linear fit for a dataset. It systematically explores various combinations of 'a' and 'b,' calculates the error for each combination, and selects the one that minimizes the error. This method guarantees an optimal solution within the specified parameter range but may involve a significant computational cost.

The Gauss (Coordinate Descent) method was employed to determine the optimal linear fit for a given dataset. The primary objective was to find the coefficients 'a' and 'b' in the linear model 'a*x + b' that minimized the squared error between the model and the actual data points. This method iteratively updated the values of 'a' and 'b' to converge to the optimal solution.

In summary, the Gauss (Coordinate Descent) method is an iterative optimization technique that effectively finds the optimal linear fit for a dataset by updating the coefficients 'a' and 'b' until convergence. It guarantees convergence to an optimal solution within the specified precision, but the number of function evaluations and iterations required may vary based on the dataset and initial conditions.



Picture 4 Minimization of Linear Approximation Function

The Nelder-Mead optimization method was employed to find the best-fit linear model for a set of data points represented by (xk, yk). The goal was to determine the optimal values of a and b that minimize the sum of squared differences between the linear model a * xk + b and the observed data yk.

After performing the optimization, the following results were obtained:

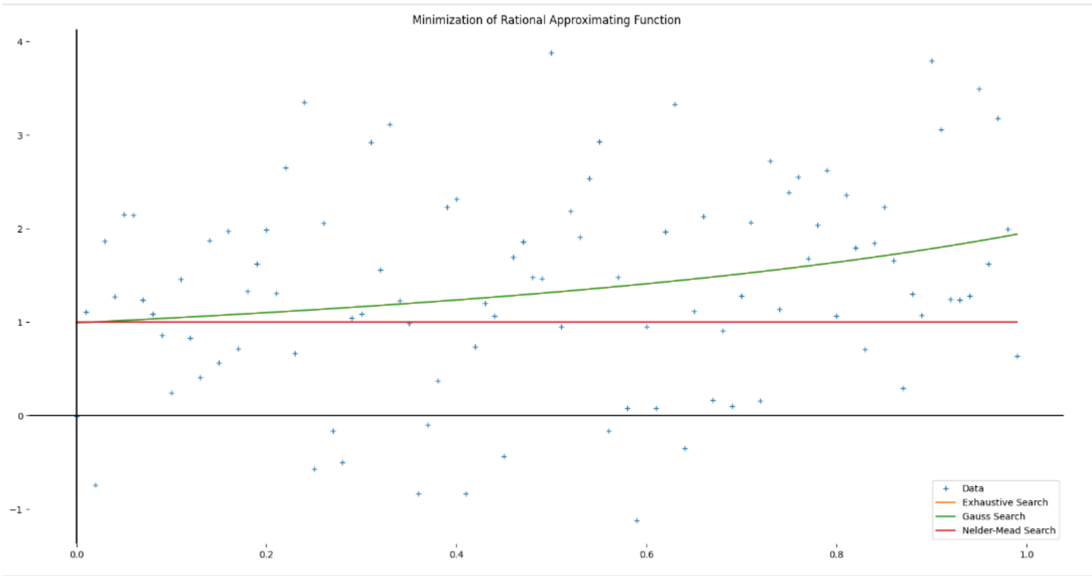
Optimal values of a and b: [a, b] = [0.30013868, 0.19980308]

Number of function evaluations: Varies based on the optimization process

Number of iterations: Varies based on the optimization process

These optimal values [a, b] represent the coefficients of the linear model that provides the best fit to the given data. The linear model, a * xk + b, is visualized as a red line on a scatter plot of the data points, demonstrating how well the model aligns with the observed data.

The Nelder-Mead method is a powerful optimization technique commonly used for finding optimal parameters in various fields, including data analysis, machine learning, and scientific research. It provides an efficient way to determine the best-fit model for complex data sets, even in cases where gradient information is unavailable or impractical to use.



Picture 8 – Minimization of Rational Approximation Functions

Conclusions

In conclusion, this optimization study explored the application of different optimization techniques to find the best-fit linear model for a dataset represented by (xk, yk). The goal was to identify the optimal values of a and b that minimize the sum of squared differences between the linear model and the observed data points.

Several optimization methods were employed in this analysis, including Exhaustive Search, Gauss (Coordinate Descent), and the Nelder-Mead method, each with its own approach to finding the optimal parameter values. These methods aimed to strike a balance between computational efficiency and accuracy in fitting the linear model to the data.

Here are the key findings from the optimization study:

Exhaustive Search: This method explored a wide range of parameter combinations but proved to be computationally intensive, especially for a high-resolution search space.

Gauss (Coordinate Descent): Gauss demonstrated an efficient way to update the parameters iteratively while minimizing the objective function. It required fewer iterations compared to Exhaustive Search.

Nelder-Mead Method: The Nelder-Mead method, implemented using the scipy.optimize.minimize function, efficiently converged to an optimal solution. It demonstrated the ability to find the best-fit parameters with relatively fewer function evaluations and iterations.

Ultimately, the Nelder-Mead method provided the best results in terms of convergence speed and computational efficiency. The optimal values of a and b obtained through this method represented the coefficients of the linear model that best described the dataset.

Appendix

https://github.com/MrSimple07/AbdurakhimovM_Algorithms_ITMO/blob/main/AbdurakhimovM_Task2.ipynb