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

*Vdovkina Sophia*

*Syrchenko Arina*

*Academic group: J4133c*

Accepted by

Dr Petr Chunaev

St. Petersburg

2022

**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.

**Formulation of the problem**

Generate random numbers and . Furthermore, generate the noisy data , where , according to the following rule:

,

where are values of a random variable with standard normal distribution. Approximate the data by the following linear and rational functions:

1. (linear approximant)
2. (rational approximation)

by means of least squares through the numerical minimization (with precision ) of the following function:

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.

**Brief theoretical part**

It is not uncommon for the mathematical problem to be able to calculate its first- and second-order derivatives. This information can be successfully utilized in solving the optimization problem.

*Gradient Descent*

Gradient descent is a first-order iterative optimization method for differentiable functions. Gradient descent is based on the observation that if the multi-variable function is defined and differentiable in a neighborhood of a point *a*, then *f(x)* decreases fastest in the direction of . It can be written down as the following formula:

Gradient descent algorithm may converge faster, if step size is chosen correctly.

The conjugate gradient is an iterative method for unconditional optimization in a multidimensional space. It works when the function is approximately quadratic near the minimum, which is the case when the function is twice differentiable at the minimum and the second derivative is non-singular there.

This method simulates such a physical quantity as momentum in order to step out of the local minimum. The conjugate gradient method is seen as an evolution of the gradient descent where instead of doing small steps towards the negative gradient, it uses an adjustable step length and performs a line search in this direction until it reaches the minimum.

*Newton’s method*

Newton's method in optimization is applied to the derivative *f ′* of a twice-differentiable function *f* to find the roots of the derivative (solutions to *f ′*(x) = 0), also known as the stationary points of . The geometric interpretation of Newton's method is that at each iteration, it amounts to the fitting of a paraboloid to the surface of *f(x)* at the trial value *xk* having the same slopes and curvature as the surface at that point, and then proceeding to the maximum or minimum of that paraboloid (in higher dimensions, this may also be a saddle point). If *f* happens to be a quadratic function, then the exact extremum is found in one step.

is a convex and twice differentiable. We should find a root of constructing a sequence from an initial approximation so that as , where

From the Taylor expansion of *f* near ,

.

We use this quadratic function (with respect to ) as an approximant to *f* in a neighbourhood of . The vertex of the corresponding parabola gives us the next point . To find the vertex x-coordinate, we write:

. Incrementing by this gives us a point closer :

It is proved that for the chosen class of *f* one has as .

*Levenberg-Marquardt algorithm*

The Levenberg-Marquardt algorithm combines two numerical minimization algorithms: the gradient descent method and the Gauss-Newton method. Levenberg-Marquardt is a popular alternative to the Gauss-Newton method of finding the minimum of a function that is a sum of squares of nonlinear functions,

Let the Jacobian of be denoted , then the Levenberg-Marquardt method searches in the direction given by the solution *p* to the equations

,

where are nonnegative scalars and *I* is the identity matrix. The method has the nice property that, for some scalar related to , the vector is the solution of the constrained subproblem of minimizing subject to .

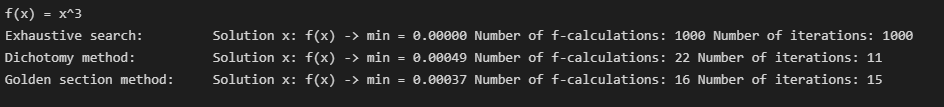
**Results**

1. We used the one-dimensional methods of exhaustive search, dichotomy and golden section search to find an approximate (with precision Ɛ = 0.001) solution : for the following functions and domains:1. 

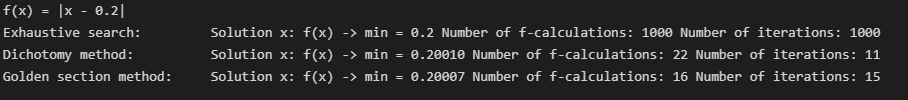
2. 

3. 

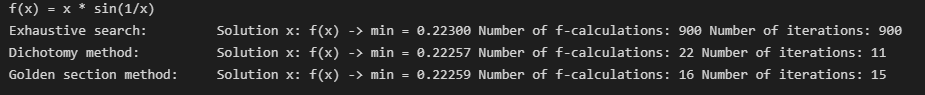
The results are shown in the pictures 1 – 3.



Picture 1 – results for the first function



Picture 2 – results for the second function

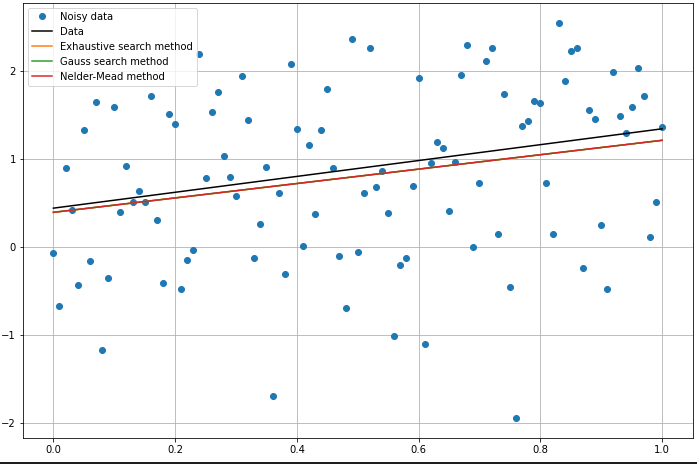


Picture 3 – results for the third function

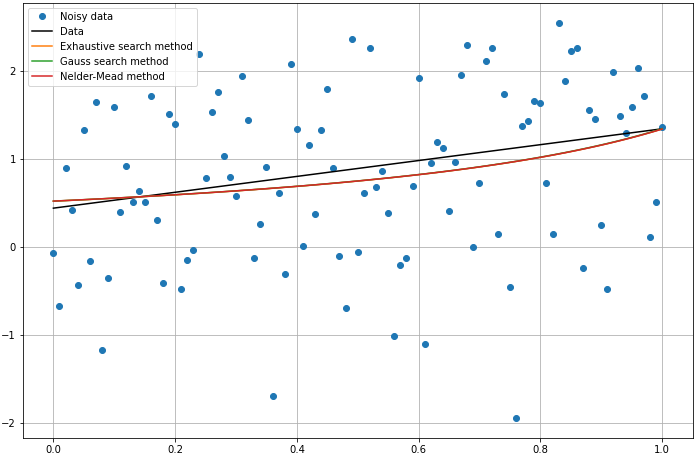
As it shown in the pictures above all methods found almost the same solution with the given accuracy. The exhaustive search method required maximum number of function calculations and maximum number of iterations. It can be explained as in exhaustive search we should iterate through all possible values.

The golden section method required fewer function calculations as at each iteration (except for the first one) the method requires only one function calculation, in contrast to the

1. Noisy data was generated and approximated by linear and rational functions by means of least squares through the numerical minimization of the given function. To solve the



Picture 4 – linear optimization



Picture 5 – rational optimization

All methods in each of the approximation functions came to the same value of the minimization function. As expected, the exhaustive method required more iterations and f-calculations than others. Also, for each approximation function, the Gauss method required fewer iterations than the Nelder-Mead method. However, the Nelder-Mead method required fewer function evaluations.

**Conclusion**

As a result, methods of unconstrained nonlinear optimization were implemented to deal with two particular problems. One-dimensional optimization was used to solve a problem of approximating the minimum of a function, we used exhaustion method, dichotomy and golden section. Dichotomy was better in minimum number of iterations and golden search was better in the minimum number of function calculations. For multidimensional (in particular, two-dimensional) optimization, a function approximation problem was considered, for which three types of minimizations were implemented for two approximation functions.

**Appendix**

Source code is available on <https://github.com/sophia-vdovkina/Analysis-and-development-of-algorithms/blob/main/Task%202/task.ipynb>