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
Kozlov Alexey
Dmitriy Koryakov
J4133c
Accepted by
Dr Petr Chunaev

St.Petersburg
2022

Table of Contents

**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

**Problems**

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:

$$y_k = \alpha x_k + \beta + \delta_k, \; x_k = \frac{k}{100}, \; (1)$$

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)
2. *F(x, a, b) =* $\frac{a}{1+bx}$ *(rational approximant)* (3)

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

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2 \; (4)$$

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**

*Gradient descent.*

Gradient descent is based on the observation that if f is differentiable at a, then f (x) decreases fastest in a neighbourhood of a in the direction of $-\nabla f(a)$. One may write down the following formula:

$$a_{n+1} = a_n - \beta \nabla f(a_n), \; \beta_n > 0, \; n = 0, 1, \ldots$$

starting with some initial approximation $a_0$.

*Conjugate Gradient method.*

Calculate the steepest direction $\Delta a_n = -\nabla f(a_n)$.

Compute βn according to certain formulas.

Update the conjugate direction $s_n = \Delta a_n + \beta_n s_{n-1}$. Find $\alpha_n = \arg\min_\alpha f(a_n + \alpha_{sn})$.

Update the position: $a_{n+1} = a_n + \alpha_n s_n$.

*Newton's method. Multidimensional case.*

Let $f : R^n \to R$ be convex and $Hf(x)$ is invertible for $x \in R^n$.

The one-dimensional scheme can be generalized to several dimensions by replacing the derivative with the gradient, $\nabla f$, and the reciprocal of the second derivative with the inverse of the Hessian matrix, Hf :

$a_{n+1} = a_n - [Hf(a_n)]^{-1} \nabla f(a_n)$, n=0,1,....

By decreasing the step size, one obtains the relaxed Newton's method:

$a_{n+1} = an - \beta[Hf(a_n)]^{-1} \nabla f(a_n)$, $\beta \in (0, 1)$, n = 0, 1, . . . ,

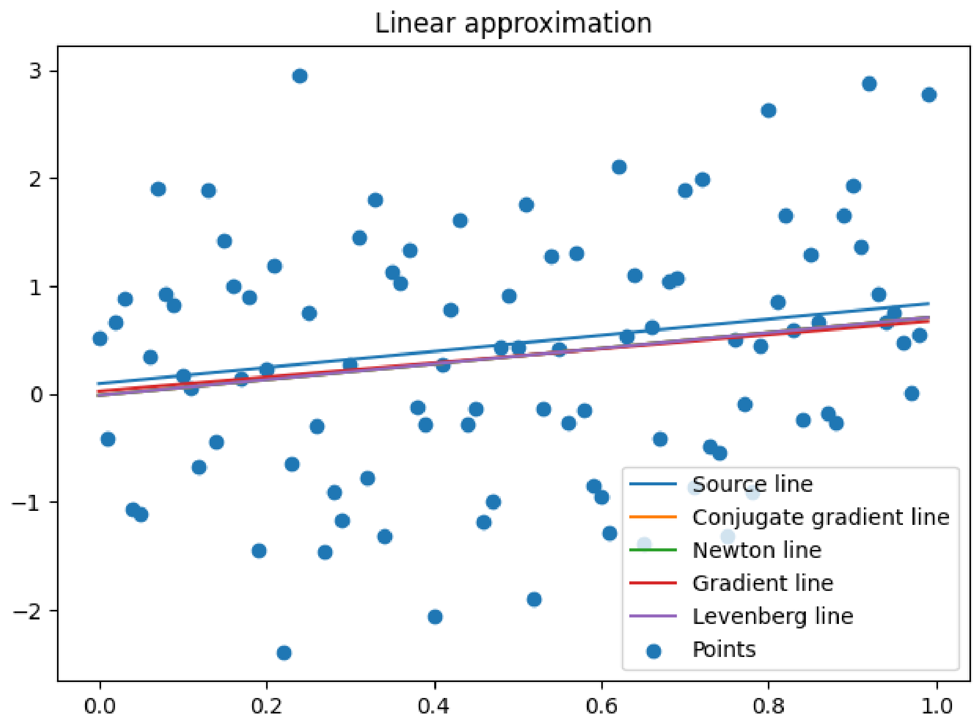
to guarantee the method's convergence. One often supposes then that f is strictly convex and Hf is Lipschitz.

*Levenberg-Marquardt algorithm (LMA).*

The LMA is a pseudo-second order method, its application is to solve non-linear least squares problems.

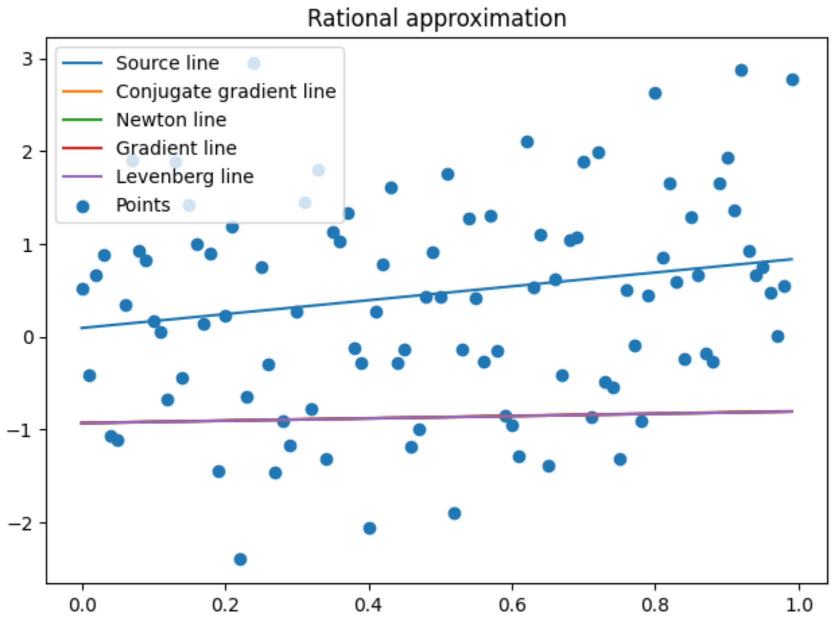
**Solution**

Linear approximation plot and table



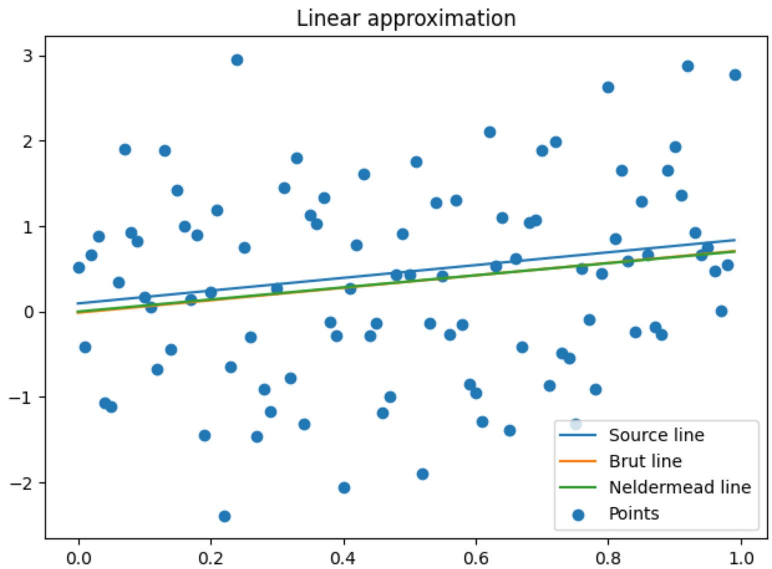

| | iterations | calculations | loss |
|---|---|---|---|
| **Brut Force** | 1000000.0 | 1000000.0 | 123.201497 |
| **Neldermead** | 55 | 104 | 123.205948 |
| **Gradient** | 154 | 308 | 123.248009 |
| **Conjugate gradient** | 2 | 15 | 123.201497 |
| **Newton** | 2 | 15 | 123.201497 |
| **Levenberg** | 159 | 159 | 123.201498 |

Rational approximation plot and table

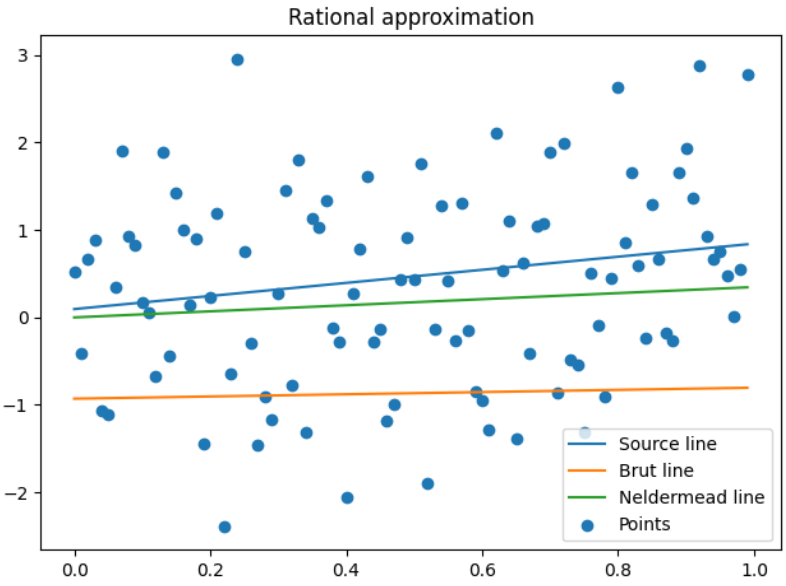



| | iterations | calculations | loss |
|---|---|---|---|
| **Brut Force** | 1000000.0 | 1000000.0 | 118.972927 |
| **Neldermead** | 39 | 62 | 127.622963 |
| **Gradient** | 128 | 256 | 118.973762 |
| **Conjugate gradient** | 13 | 195 | 118.972926 |
| **Newton** | 12 | 78 | 118.972926 |
| **Levenberg** | 161 | 161 | 118.972927 |

Linear approximation plot from Task 2




Rational approximation plot form Task 2




## Conclusions

With a linear approximation, all methods have the same loss value. With a rational approximation, Neldermead has the highest loss value, the rest are equal in losses to each other. Conjugate Gradient and Newton's method is the most optimal in terms of the ratio of the value of losses and calculations.

## Appendix

Link to github - https://github.com/OnlyOneUseAcc/algorithms-RD-practise