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

Cheng-Yuan and Ma

Academic group

J4133c

Petr Chunaev *St.*

Petersburg 2022

Task# 2

Language : python

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

Problem : 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:

I.

Theory:

exhaustive search : In this approach, we generate each element in the problem and then select the ones that satisfy all the constraints, and finally find a desired element

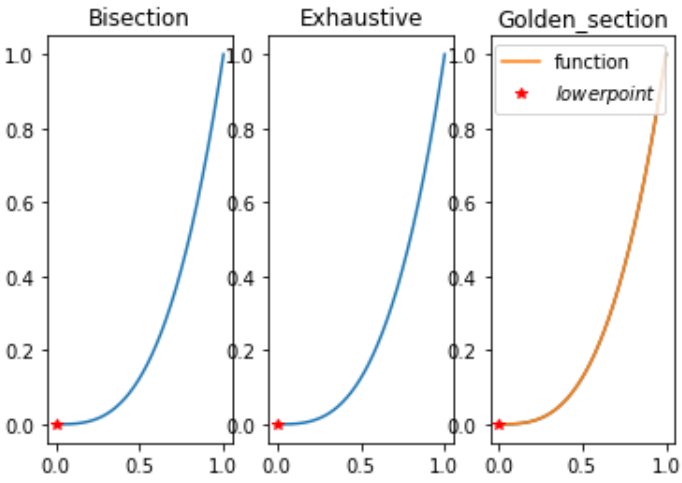
dichotomy search : use bound index to keep finding mid index , and keep trying find the smallest func(index)

golden section search : use golden rate (0.618) and two bound indexes to shrink the range and find the min

Gauss (coordinate descent) : is used to solve non-linear least squares problems, which is equivalent to minimizing a sum of squared function values

Nelder-Mead : is a numerical method used to find the minimum or maximum of an objective function by reflection, expansion, contraction and shrink coefficients in a multidimensional space

1. $f(x)=x^{\wedge}3, x\in[0,1]$;



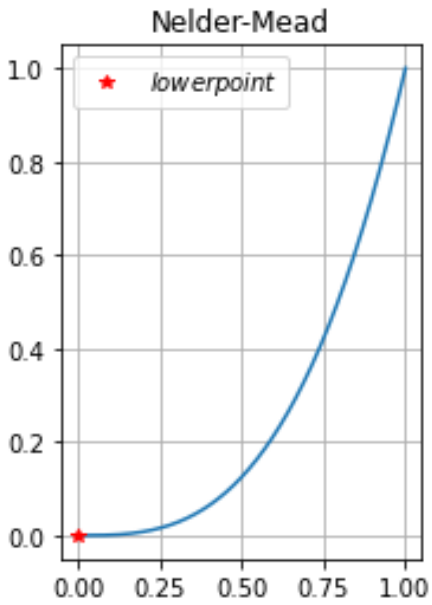

```
iteration is  : 10
The value of root is :  0.0010
evaluation of objective function is : 41


bounday is minimun
evaluation of objective function is : 2995
iteration is  : 998


evaluation of objective function is : 32
iteration is  : 15
```

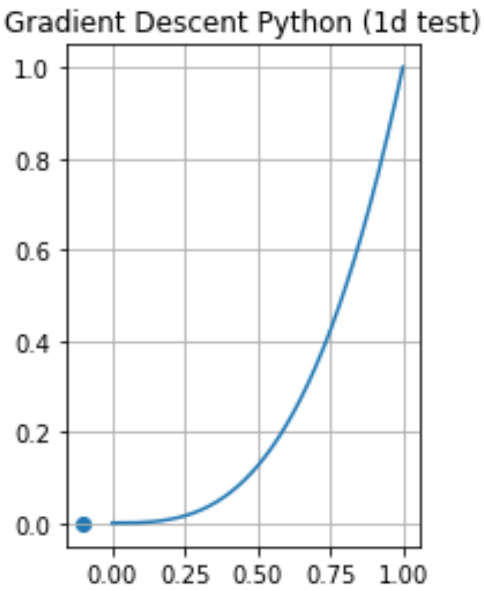
(iter , evalation)

Bisection (10,41)

Exhaustive (998,2995)

golden_sec(15,32)





\

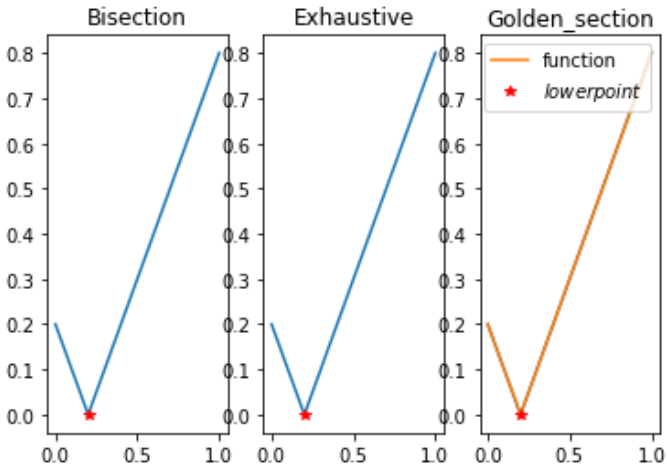
```
  success: True
        x: array([1.e-06])
iteration is : 1
evaluation of objective function is : 2


iteration is  : 1
lowest point -0.099999,-0.001000
evaluation of objective function is : 1
```

(iter , evalation)

Nelder:(1,2)

Gradient:(1,1)

2. $f(x)=|x-0.2|, x\in[0,1]$;



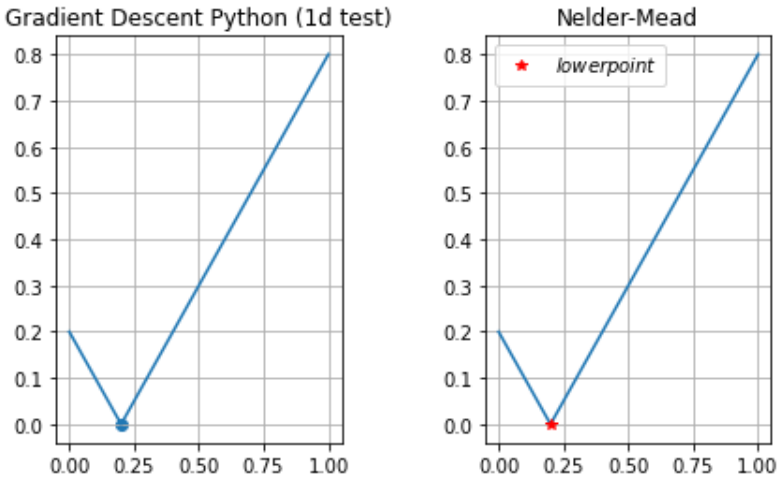
(iter , evalation)

Bisection (10,25)

Exhaustive (200,601)

golden_sec(15,18)

```
iteration is  : 10
The value of root is :  0.2002
evaluation of objective function is : 25


The minimum point lies between  0.19900080100000025  &  0.20100079900000026
evaluation of objective function is : 601
iteration is  : 200


evaluation of objective function is : 18
iteration is  : 15
```



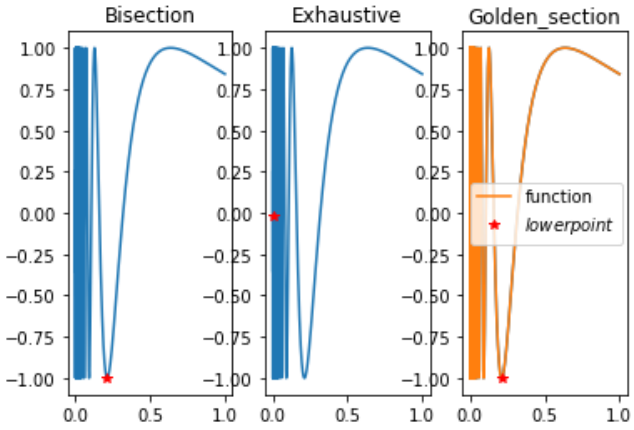

```
iteration is : 2
evaluation of objective function is : 72
nelder is called : 1


iteration is  : 52
lowest point 0.199165,0.000835
evaluation of objective function is : 52
```

(iter , evalation)

Nelder:(1,2)

Gradient:(52,52)

3. $f(x)=xsin(1/x), x\in[0.01,1]$. :



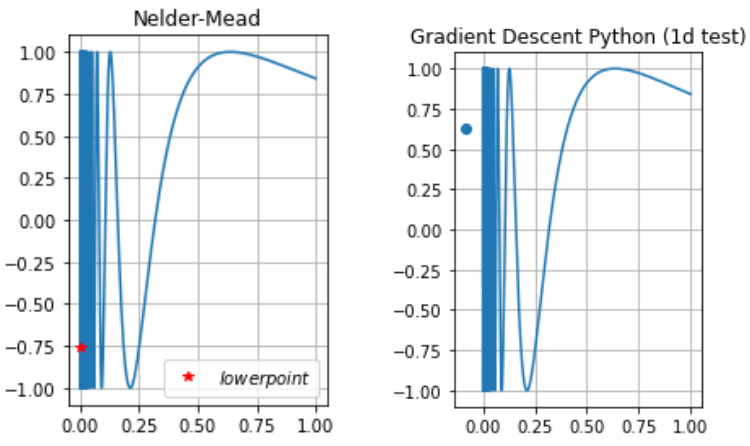

```
iteration is  : 10
The value of root is :  0.2119
evaluation of objective function is : 17


The minimum point lies between  0.0010009989999999998  &  0.003000996999999999
evaluation of objective function is : 7
iteration is  : 2


evaluation of objective function is : 18
iteration is  : 15
```

(iter , evalation)

Bisection (10,17)

Exhaustive (2,7)

golden_sec(18,15)




```
iteration is : 1
evaluation of objective function is : 97


iteration is  : 1
lowest point -0.084146,0.630563
evaluation of objective function is : 1
```

(iter , evalation)

Nelder:(1,97)

Gradient:(1,1)

II.

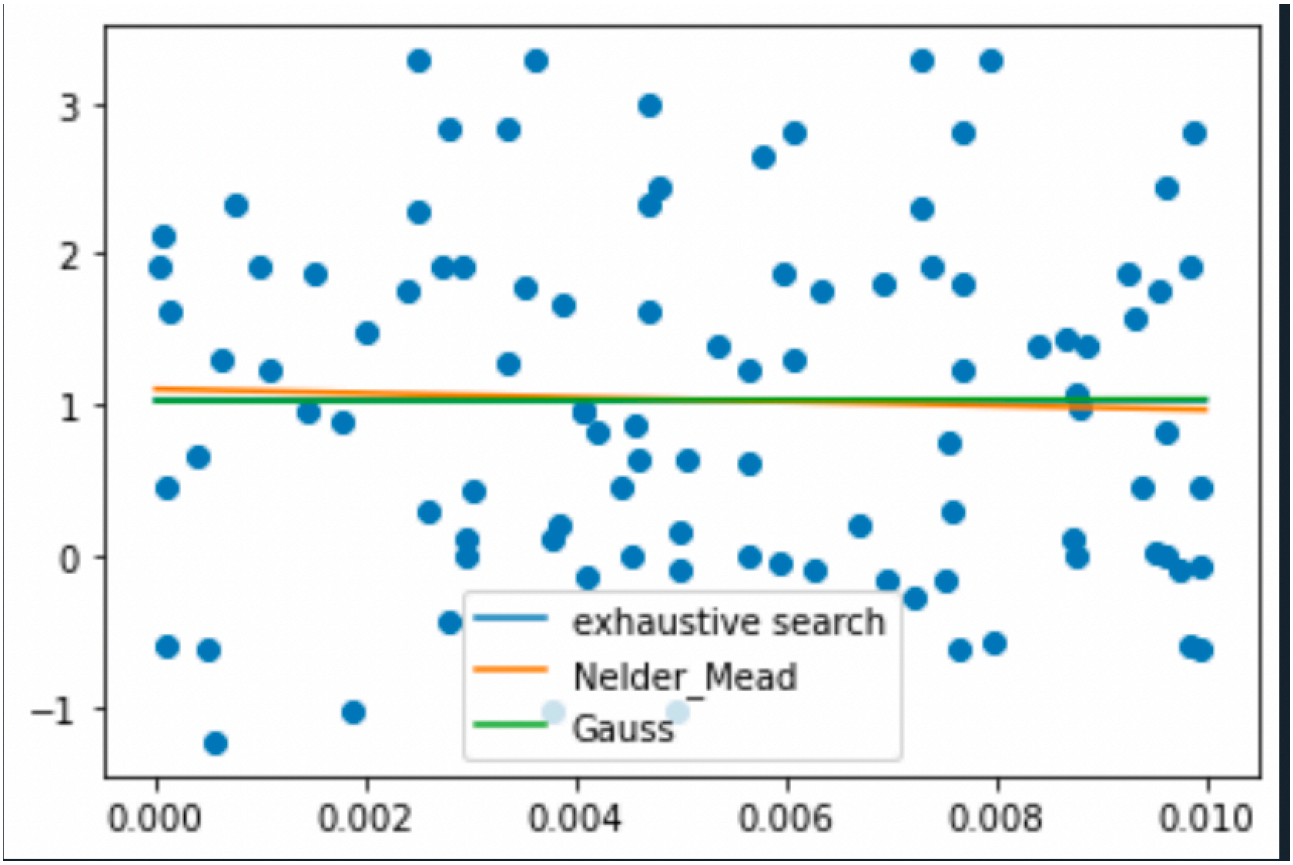
Generate random numbers $\alpha \in (0,1)$ *and* $\beta \in (0,1)$. *Furthermore, generate the noisy data* $\{x_>, y_>\}$, *where* $k = 0, \ldots, 100$, *according to the following rule:*

$y_k = \alpha x_k + \beta + \delta_k, \quad x_k = k/100$

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:*

```
exhaustive search : iteration=1000 ,cost=1.2873135027003073 ,evaluation of function=1000
Nelder_Mead  :initial_node=[1, 1] , iteration=46 , cost=1.285652167182753  ,evalutaion of function=86
Gauss  :initial_node=[1, 1] , iteration=1 , cost=1.2875700103737664  ,evalutaion of function=9
exhaustive search with rational approximant : iteration=1000 ,cost=1.4276755342028198 ,evaluation of function=1000
Nelder_Mead with rational approximant :initial_node=[1, 1] , iteration=51 , cost=1.2858765098478144  ,evalutaion of function=97
Gauss with rational approximant  :initial_node=[1, 1] , iteration=1 , cost=1.2870772481722241  ,evalutaion of function=9
```

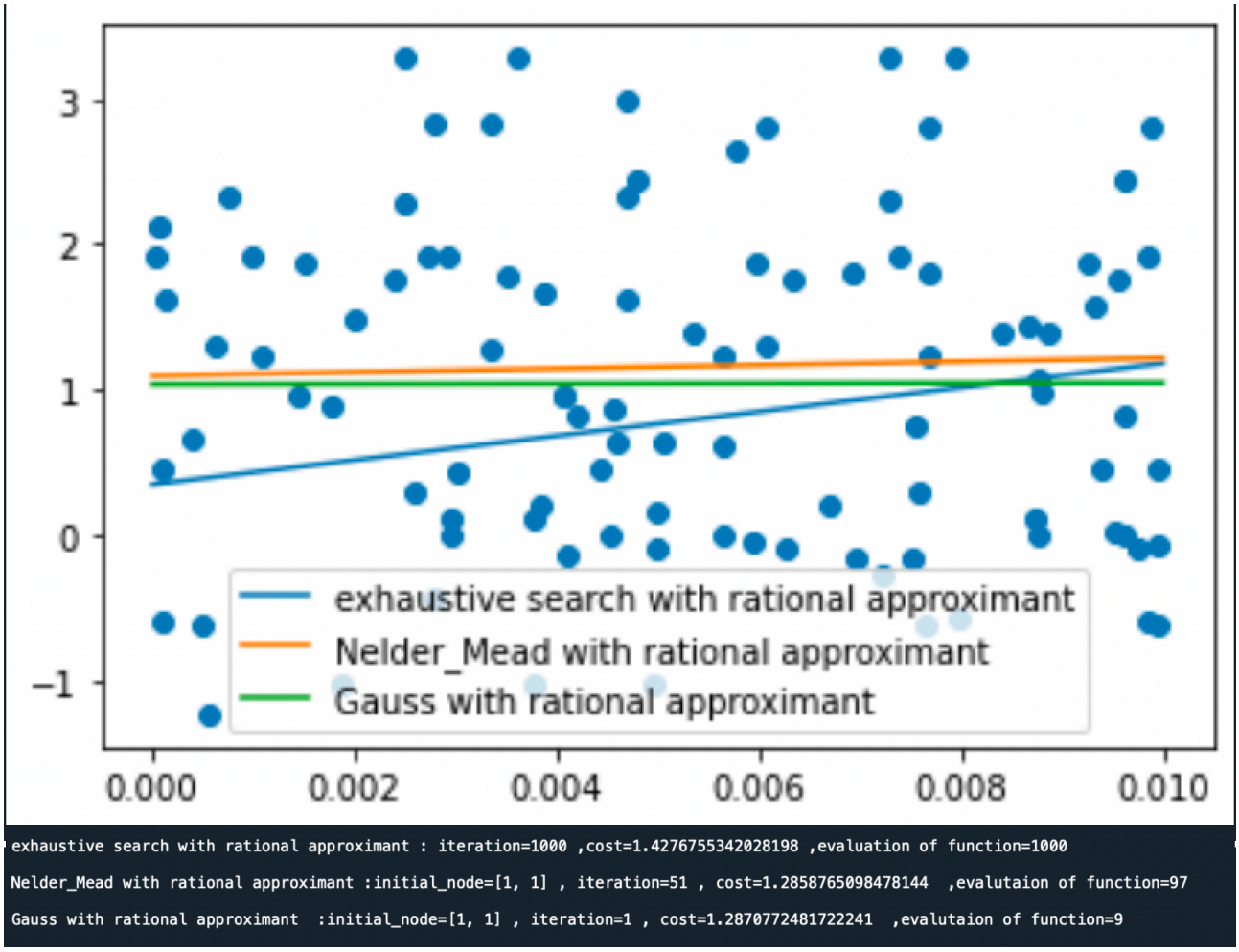
1. $F(x,a,b)=ax+b$*(linear approximant)*




```
exhaustive search : iteration=1000 ,cost=1.2873135027003073 ,evaluation of function=1000
Nelder_Mead  :initial_node=[1, 1] , iteration=46 , cost=1.285652167182753  ,evalutaion of function=86
Gauss  :initial_node=[1, 1] , iteration=1 , cost=1.2875700103737664  ,evalutaion of function=9
```

2. $F(x, a, b) = a/1+bx$ *(rational approximant),*

by means of least squares through the numerical minimization (with precision ε = 0.001)



Conclusion :

In first section , through graphs which I created , we can see that in most of formulas , we can achieve minimum values from each algorithm which I tested . But as we can see I found difficulty to use gauss and Nelder-Mead to find minimum from any function with trigonometric .

Most of time it cost a lot of iterations for exhaustive search to find a min , but somehow in third one with trigonometric , it only cost 2 times of iterations and 7 times of function , probably there is error in my algorithm . But we still can conclude that exhaustive search is not efficient to make calculation , it takes too much time to calculate each possibility .

On the contrary , we can see that Nelder-Mead and gauss show a very efficient way to get to result . But there is little mistake , so I can not get to desired answer .

In the second part, we try to approximate the data using linear and rational approximations. As we can see, when we use linear approximation, the lines just converge to almost the same line, conversely, rational approximation does not converge as well as linear approximation. Of the three methods we used in this task, we can see that the exhaustive method requires more iteration time to come to a conclusion, followed by the nelder mead, whereas with the Gaussian method, it requires much fewer iterations, This means it is much more efficient than the other two ways.

Appendix

https://github.com/MaChengYuan/task2.git