FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 3

“Algorithms for unconstrained nonlinear optimization. First- and secondorder methods”

Performed by

Pakulev Aleksandr

J4133c

Accepted by

Dr Petr Chunaev

St. Petersburg

2021

Brief information:

In this task need to implement some gradient descent algorithms.

Bellow brief algorithms descriptions:

1) Gradient descent –fast gradient descent.

   Point of this algorithm type is that on each iteration you must found the least L, which corresponded minimum point on current gradient vector.

2) Conjugate Gradient Descent – on each step we optimize not sum of gradients, but one random axes.

3) Newton's method – iterative method, where at each step we found x_n as f(x_n-1) / f'(x_n-1), or as f'(x_n-1) / f''(x_n-1)

4) Levenberg-Marquardt algorithm – algorithms which optimize not all functions. It optimizes set of functions use the least squares method.

In task must find approximated line coefficients, use methods above:

(y = a * x + b) a and b coefficients.

Data set must be generated with according formula and additional noise based on normal distribution.

Here must compare two different approximant functions.

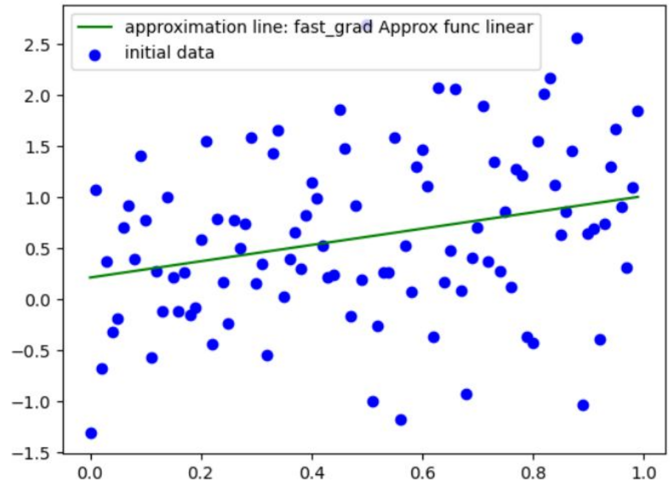
First – linear function y = a * x + b

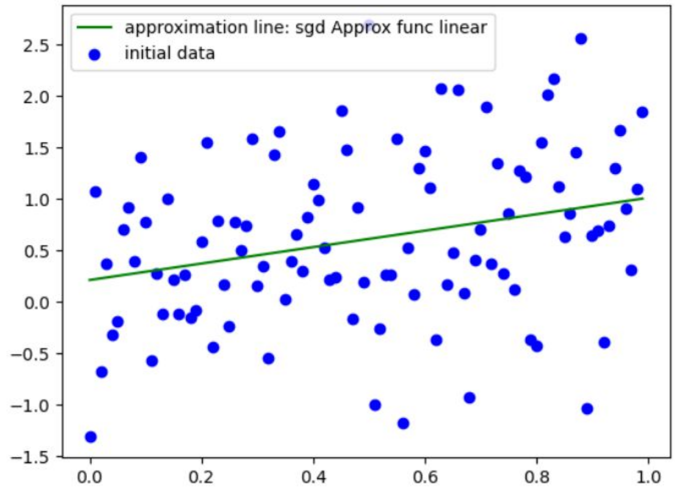
Second – rational function y = a / (1 + x * b)

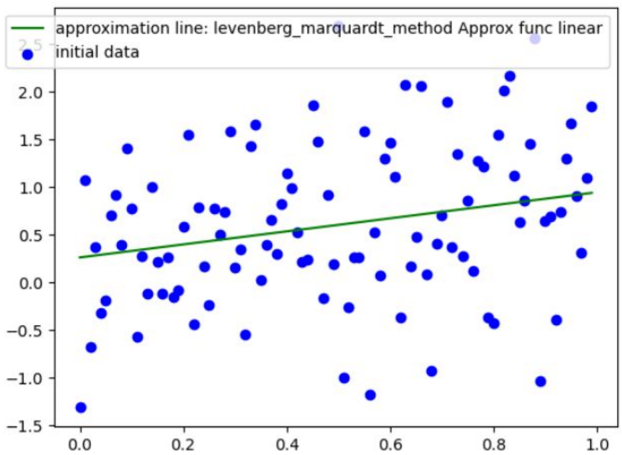
For locate the best approximate parameters must use method of least squares. In simple terms: required found such coefficient a and b in which D(a, b) = sum(F(x_k, a, b) – y_k) ^ 2, k = 0…100 the least. Function F(x, a, b) described above.

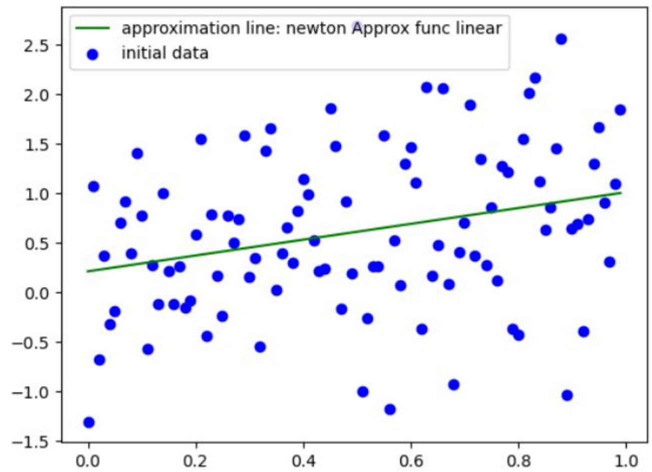
Result:

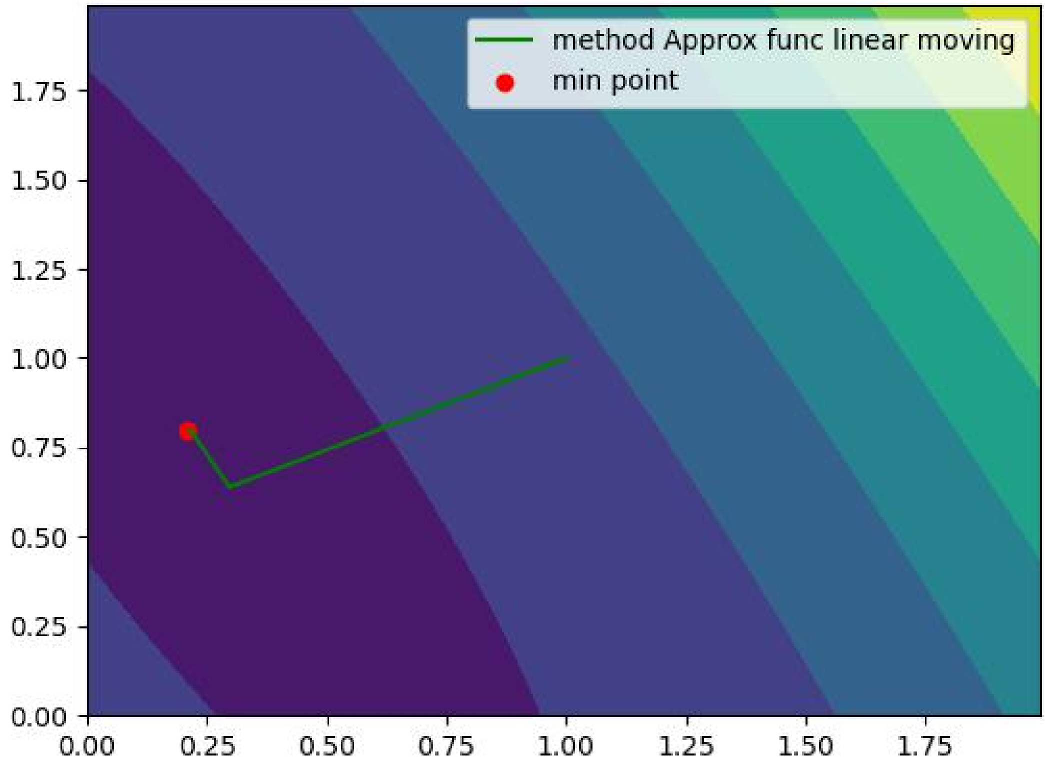
Results for **linear approximations** with “depth” map for fast gradient descent.







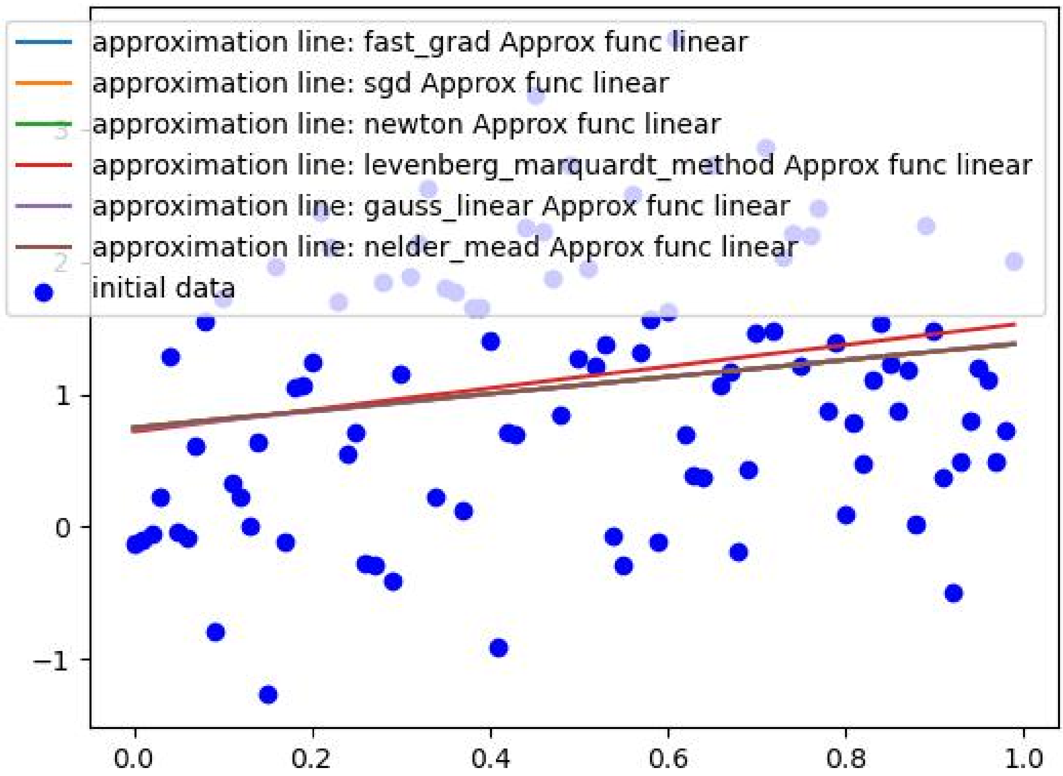




Green line is gradient moving, red point is minimum point.

Method name	Iterations count	Min value
Fast gradient descent	4	63.044057878195574
Exhaustive search	100_000	63.044686801804346
Conjugate gradient descent	2	63.044056
Newton method	3	63.04405631653349
Levenberg marquardt method	Not reported	63.157737225684656
Gauss linear	18	63.04640389294469
Nelder-Mead	65	63.044056332797574

Table above contains method for linear approximation, with iteration count and minimum value, that may reach methods from this task and previous task.

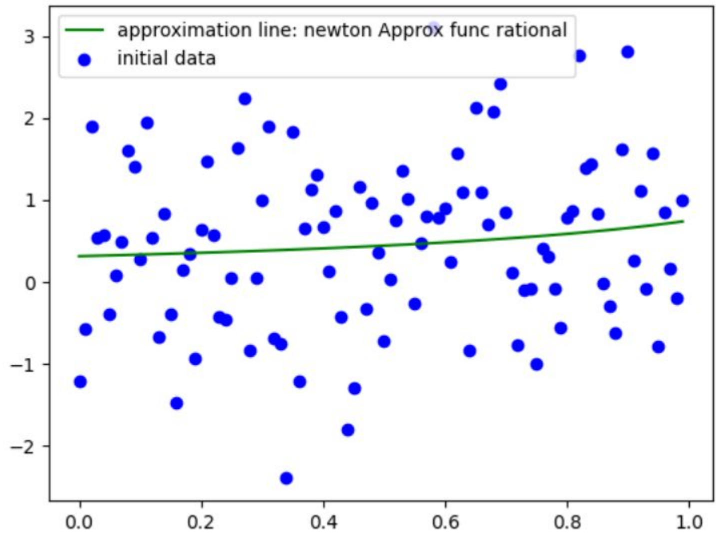


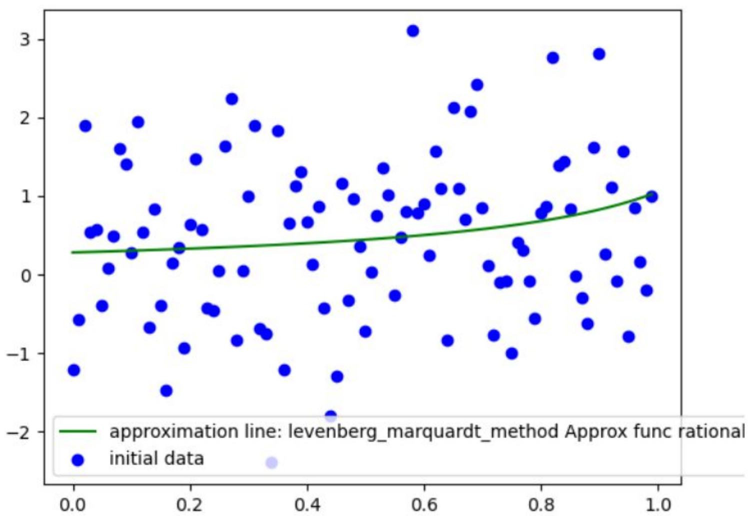
Here all plots on one line.

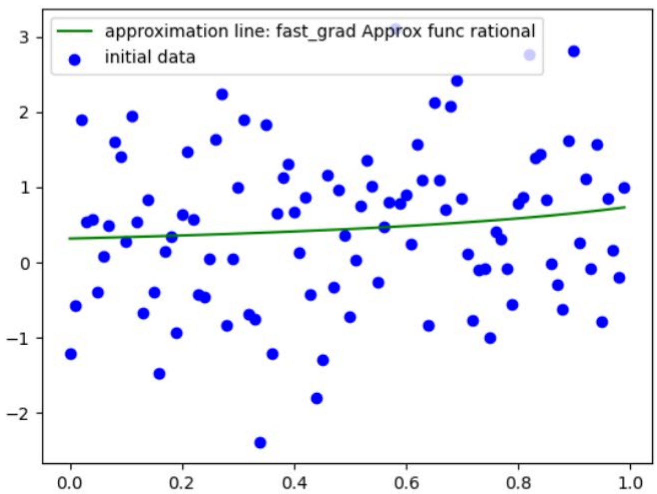
As you can see all methods except Levenberg gives same result. I suppose it's linked with implementation detail of scipy.

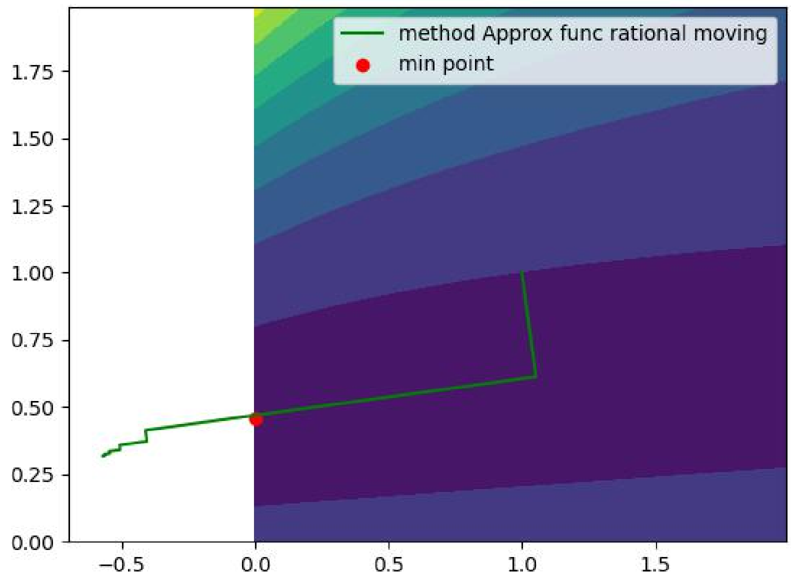
Rational approximation.

Here additional will be introduced points on which such value reached. It’s done because this method best result usually achieves when one of parameter is negative.









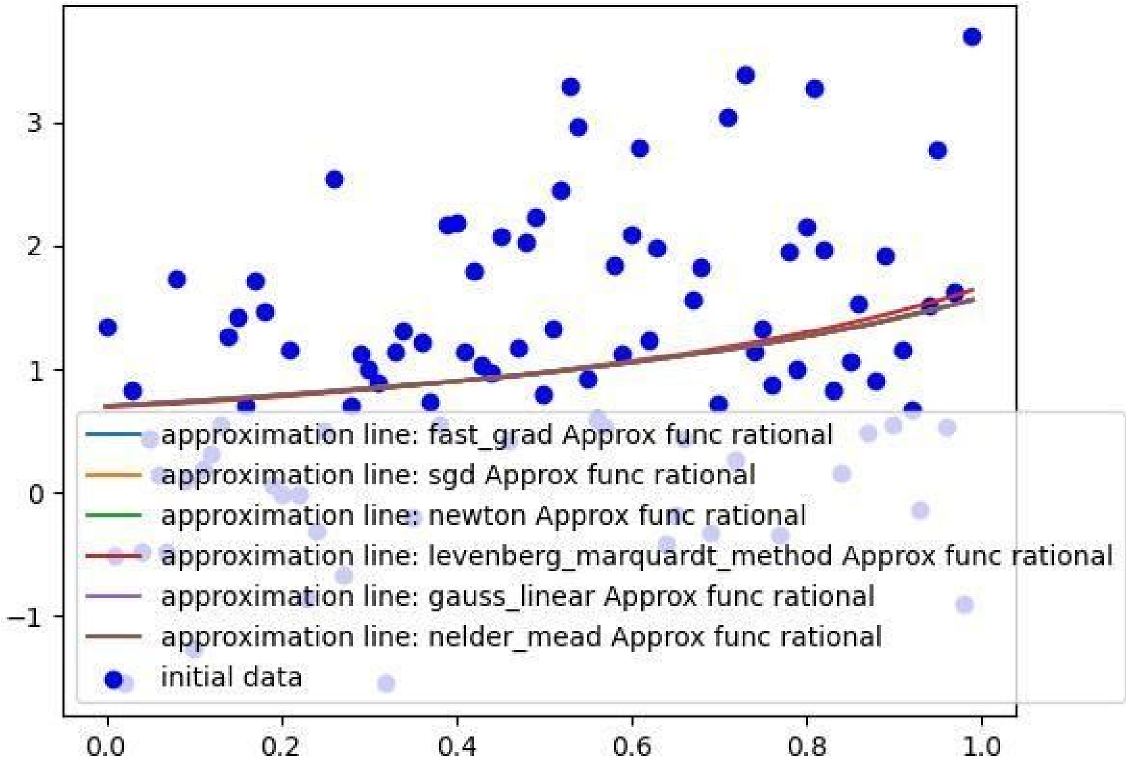
Here fast gradient descent moves to negative plot part, because over there achieve the best result. But for exhaustive search explored area only in 0-1 x 0-1 cell.

Min point here is min point that found exhaustive search.

Textual result.

Method name	Iterations count	Min value	Min point
Fast gradient descent	12	106.85750778543712	(0.3183956670, -0.5707627673)
Exhaustive search	100_000	108.62759186980536	(0.46, 0.0)
Conjugate gradient descent	12	106.856553	(0.31531249, -0.57899581)
Newton method	12	106.85655278677852	(0.31531173, -0.57899785)
Levenberg marquardt method	Not reported	107.751832027106	(0.28153272, -0.73243139)
Gauss linear	25	106.85826719499144	(0.320194414, -0.569810780)
Nelder-Mead	62	106.85655280280342	(0.31531907, -0.5790002)

As can consider from sheet above for rational function better result often achieve when parameter b is negative.



Conclusion:

As result, it may be noticed, that first and second order function coverage faster (required less operations) then methods which does not use gradients.

Additional conclusion -- rational function in current terms achieve better result when parameter b is about zero or less than zero.