Lab work 1

October 19, 2020

Instructor: Djaffar Boussaa (boussaa@lma.cnrs-mrs.fr)

Please follow the following instructions:

1. use separate files for the objective values, gradients and second derivatives for each of the test functions;
2. write your code for a general, not a specific, objective. The objectives, gradients and second derivatives will be passed as arguments to your code;
3. use separate files for linesearches.

Algorithm implementation

1. Implement the steepest descent method with Backtracking Armijo linesearch.
2. Implement the Newton method direction Backtracking Armijo linesearch.
3. Implement the BFGS method with weak Wolfe linesearch.

Test functions

1. Test your code on $f(x_1, x_2) = \frac{1}{2}(Mx_1^2 + x_2^2)$, where $M > 0$.
2. Test your code on Rosenbrock's function, $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$.

To go further

You may want

1. to compare your codes with Python's built-in optimization functions (see scipy.optimize) in terms of solutions, number of function evaluations, number of gradient evaluations, and number of Hessian evaluations.

2. to test your codes on the generalized Rosenbrock function.

3. to apply your codes to some of the (low-dimensional) standard test functions described in `https://en.wikipedia.org/wiki/Test_functions_for_optimization`.