Numerical optimization for large scale problems
Unconstrained optimization

Problem list, unconstrained optimization
a.y. 2022/23

Note. Whatever is your choice, I expect you to submit **a single pdf file**. The file should be submitted through the `exercise.polito.it` page from which you downloaded the assignment. If you work in a group (max 3 people) please upload the file **only once** but clearly state in the file name the family names of all team-mates.

The document is expected to report an introductory analysis of the problem, tables and/or figures summarizing your results and comments on your results. Please use **captions** in order to explain what every table and/or figure is reporting, and quote it also in the text (e.g., "In Figure xx we report the plot of....", "In Table yy we compare ...").

In general you are expected to test your solvers on some common problems, with different values of some parameters and possibly different starting points. In all the cases you should compare the results obtained, for example in terms of number of iterations and computing time, commenting your results also in view of the values of the parameters used and of the theory.

As an **appendix**, please add the commented scripts/functions you implemented in your favorite programming language. Please make sure to use sensible names for the variables and functions, and to provide enough comments and explanations to render the code readable to a non expert of the specific language.

Assignment:

Implement at least two of the following numerical methods for unconstrained optimization:

1. nonlinear conjugate gradient method (either Fletcher and Reeves or Polak-Ribière)
2. Newton method
3. Newton method with finite differences
4. Inexact Newton method
5. Nelder-Mead
6. steepest descent method

Test them on the Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

with starting point $x_0 = (1.2, 1.2)$ and $x_0 = (-1.2, 1)$.

In all the cases (except for Nelder-Mead), complement the method with a backtracking strategy for the line search, imposing sufficent decrease condition with parameters $\rho = 0.5$ and $c = 10^{-4}$. Try to tune the parameters, if it is not working well.

Apply the codes to at least three test problems taken from

`https://www.researchgate.net/publication/325314497_Test_Problems_for_Unconstrained_Optimization`

using several starting points, and using **large** values of the dimension n (i.e., $n = 10^d$, d at least 3 or 4).

Realize a throughout comparison among the codes. The comparison should be performed comparing, for each test problem: number of failures/successfull runs, number of iterations to satisfy a fixed stopping criterion (in case of inexact Newton: inner and outer iterations), computational cost... You may possibly report in a table some data for an overview of the behaviour of the two methods. Comment your results also in view of the expected theoretical behaviour.