Numerical Optimization for Large Scale Problems
Assignments

Politecnico di Torino, A.Y. 2024/2025

PLEASE, CAREFULLY READ ALL THE INSTRUCTIONS BEFORE PREPARING AND SUBMITTING YOUR REPORT

Recall that the assignment allows you to **score up to 12 points**, according both to the report contents and defense. Suitably addressing all the points reported, allows you to score the corresponding points. Each team has to choose a single assignment either from the Unconstrained or the Constrained optimization section.

1 Assignment on Unconstrained Optimization

1. Implement **exactly two** out of the following numerical methods for unconstrained optimization:

   1.1. [**2 points**] Nelder-Mead

   1.2. [**0.5 points**] Steepest descent method

   1.3. [**1 point**] Nonlinear conjugate gradient method (either Fletcher-Reeves or Polak-Ribière)

   1.4. [**2 points**] Modified Newton method

   1.5. [**0.5 points**] Inexact Newton method[1]

   1.6. [**2 points**] Truncated Newton method

   In all the cases (except for Nelder-Mead), embed the method with a back-tracking strategy for the line search.

2. [**0.5 point**] Test your implementations on the Rosenbrock function

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

   with starting point $\boldsymbol{x}^{(0)} = (1.2, 1.2)$ and $\boldsymbol{x}^{(0)} = (-1.2, 1)$, reporting the behavior both with tables (see item 3, below) and figures.

   Impose sufficient decrease condition for the backtracking strategy; e.g., by using parameters $\rho = 0.5$ and $c = 10^{-4}$. Nonetheless, tune the parameters, if they are not working well.

3. [**7.5 points**] Apply the codes to **exactly three** test problems taken from [1] following these instructions:

   - set a random seed equal to the minimum student ID of the team members; e.g., given three team members with student IDs 9876, 1234, and 2345, the random seed must be set equal to $1234 = \min\{9876, 1234, 2345\}$.

[1] See Recommendation 3.1.

Generazione dei punti fatta!

- study the problem for these values for the dimension: $n = 10^d$, $d = 3, 4, 5$.
- use $\boldsymbol{x}^{(0)} = \bar{\boldsymbol{x}} \in \mathbb{R}^n$ as starting point, where $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ is the starting point suggested in [1] for the problem. Then, use 10 new starting points randomly generated with uniform distribution in a hyper-cube $[\bar{x}_1 - 1, \bar{x}_1 + 1] \times \cdots \times [\bar{x}_n - 1, \bar{x}_n + 1] \subset \mathbb{R}^n$.
- impose sufficient decrease condition for the backtracking strategy; e.g., by using parameters $\rho = 0.5$ and $c = 10^{-4}$. Nonetheless, tune the parameters, if they are not working well.
- If possible, apply the codes both for the case with exact derivatives and for the case where derivatives are computed by using finite differences[2] with respect to the following values for the increment h for each differentiation:

  $$h = 10^{-k}, \qquad k = 2, 4, 6, 8, 10, 12\,.$$

  Moreover, test the finite differences also using a specific increment h_i when differentiating with respect to the variable x_i, according to these values:

  $$h_i = 10^{-k}|\hat{x}_i|, \qquad k = 2, 4, 6, 8, 10, 12\,, \quad i = 1, \ldots, n\,,$$

  where $\hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_n) \in \mathbb{R}^n$ is the point at which the derivatives have to be approximated.

After running the optimization methods according to the instructions above, realize a throughout comparison among the methods, reporting data in tables and/or figures. Comment on your results also in view of the expected theoretical behavior. The comparison should be performed comparing, for each test problem:

- number of failures/successful runs;
- number of iterations to satisfy a fixed stopping criterion;
- experimental rate of convergence;
- execution time;
- whatever else you think may be useful to understand the behavior of the methods.

Recommendations:

3.1. in case of inexact Newton method: use at least two different forcing terms which are expected to yield different convergence rates, and report the number of both inner and outer iterations;

3.2. Whenever you use an iterative solver for solving linear systems, report results both with preconditioning and without preconditioning.

3.3. **Do not** use a general purpose finite-difference function, but rather be sure to exploit the structure of the function, to implement the finite difference, or your code will unlikely run in a reasonable time when applied to large dimensions (e.g., $n = 10^5$).

2 Assignments on Constrained Optimization

2.1 Assignment: Projected gradient method

1. Implement the projected gradient method and use it for performing either item (2) or (3). In particular, implement **at least one** of the following methods:

   1.1. [**0.5 points**] Projected gradient with line search along the feasible direction;

[2] See Recommendation 3.3.

1.2. [**2 points**] Projected gradient with line search along the projection arc;

2. [**7 points**] Consider the problem described by equation (3) in [2]. Use your implementation(s) of the projected gradient method to solve the problem, following these instructions:

   - set a random seed equal to the minimum student ID of the team members; e.g., given three team members with student IDs 9876, 1234, and 2345, the random seed must be set equal to $1234 = \min\{9876, 1234, 2345\}$.
   - study the problem for these values for the dimension and these intervals for the domain: $n = 2, 10^3, 10^4, 10^5$ and $X = [-5.12,\ 5.12]^n$, $[1,\ 5.12]^n$, $[-5.12,\ 5.12] \times [1,\ 5.12]^{n-1}$, $[-5.12,\ 5.12]^{n/2} \times [1,\ 5.12]^{n/2}$.
   - use $\boldsymbol{x}^{(0)} = (5.12, \ldots, 5.12) \in \mathbb{R}^n$ as starting point. Then, use 10 new starting points randomly generated with uniform distribution in a hyper-cube $[4.12, 6.12]^n \subset \mathbb{R}^n$.
   - For the case $n = 2$, visualize the objective function, the domains X, and report the behavior the methods both with tables and figures (see below, end of item 2).
   - impose sufficient decrease condition for the backtracking strategy; e.g., by using parameters $\rho = 0.5$ and $c = 10^{-4}$. Nonetheless, tune the parameters, if they are not working well.
   - Apply the codes both for the case with exact derivatives and for the case where derivatives are computed by using finite differences[3] with respect to the following values for the increment h:
   $$h = 10^{-k}, \qquad k = 2, 4, 6, 8, 10, 12\,.$$
   Moreover, test the finite differences also using a specific increment h_i when differentiating with respect to the variable x_i, according to these values:
   $$h_i = 10^{-k}|\hat{x}_i|, \qquad k = 2, 4, 6, 8, 10, 12\,, \quad i = 1, \ldots, n\,,$$
   where $\hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_n) \in \mathbb{R}^n$ is the point at which the derivatives have to be approximated.

   After running the optimization methods, according to the instructions above, realize a throughout comparison among the methods, reporting data in tables and/or figures. Comment on your results also in view of the expected theoretical behavior. The comparison should be performed comparing, for each method:

   - number of failures/successful runs;
   - number of iterations to satisfy a fixed stopping criterion;
   - experimental rate of convergence;
   - execution time;
   - whatever else you think may be useful to understand the behavior of the methods.

   Recommendations:

   2.1. **Do not** use a general purpose finite-difference function, but rather be sure to exploit the structure of the function, to implement the finite difference, or your code will unlikely run in a reasonable time when applied to large dimensions (e.g., $n = 10^5$)

3. [**9.5 points**] As the item (2) but with the problem (6)[4] in [2] and:

[3] See Recommendation 2.1.

[4] Attention, there is a typo in [2], eq. (5); i.e., the 2-dimensional version of (6). The "+" before $(y - \pi)^2$ should be a "$-$".

- study the problem for these values for the domain: $X = B(\mathbf{0}, 2\pi)$, $B((1.5-\pi)\boldsymbol{e}, (\pi + 0.5))$, $[-2\pi, 2\pi] \times [-2\pi, 1]^{n-1}$, $[-2\pi, 2\pi]^{n/2} \times [-2\pi, 1]^{n/2}$. Where $B(\boldsymbol{c}, r)$ denotes the n-dimensional ball with center $\boldsymbol{c} \in \mathbb{R}^n$ and radius $r > 0$; the vector $\boldsymbol{e} \in \mathbb{R}^n$ denotes the vector of all ones.
- use $\boldsymbol{x}^{(0)} = (-2\pi, \ldots, -2\pi) \in \mathbb{R}^n$ as starting point. Then, use 10 new starting points randomly generated with uniform distribution in a hyper-cube $[-2\pi-1, -2\pi+1]^n \subset \mathbb{R}^n$.

2.2 Assignment: Equality Constrained Quadratic Programming

1. [**2 points**] Consider and implement the quadratic problem

$$
\begin{array}{rc}
\min\limits_{x \in \mathbb{R}^n} & \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n-1} x_i x_{i+1} + \sum_{i=1}^{n} x_i \\
\text{s.t.} & \text{the sum } x_1 + x_{1+K} + x_{1+2K} + \ldots \text{ should be } \epsilon \\
 & \text{the sum } x_2 + x_{2+K} + x_{2+2K} + \ldots \text{ should be } \epsilon \\
 & \vdots \\
 & \text{the sum } x_K + x_{2K} + x_{3K} + \ldots \text{ should be } \epsilon
\end{array}
$$

where $\epsilon \in (0, 1)$ is a random value generated with uniform distribution in $(0, 1)$. Generate ϵ after setting a random seed equal to the minimum student ID of the team members; e.g., given three team members with student IDs 9876, 1234, and 2345, the random seed must be set equal to $1234 = \min\{9876, 1234, 2345\}$.

2. [**10 points**] Solve the problem above following these instructions:
   - study the problem for these values: $n = 2, 2 \cdot 10^3, 2 \cdot 10^4, 2 \cdot 10^5$ with corresponding values of K that are $K = 1, 100, 500, 1000$, respectively.
   - solve the KKT conditions with the following strategies
     - full solution of the KKT system with direct solvers (whenever possible, i.e. no memory fault)
     - full solution of the KKT with a suitable iterative solver, both without and with preconditioning.
     - Schur complement approach (both with and without having the inverse of Q).
     - Null space method.

     Recommendation: Whenever you use an iterative solver for solving linear systems, report results both with preconditioning and without preconditioning.
   - For the case $n = 2$, $K = 1$, visualize the quadratic function, the constraints, and the solutions identified by all the methods.

3 Guidelines for writing the report

1. The document is expected to report:
   1.1. an introductory analysis of the problem
   1.2. a brief description of the methods
   1.3. tables and/or figures summarizing your results
   1.4. comments on your results.
2. You must write the **name, family name, and student ID of each team member** at the beginning of the report.

3. 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 ...").

4. 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 the number of iterations and computing time, commenting on your results also in view of the values of the parameters used and of the theory.

5. As an **appendix** of the report, you must **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.

4 Submission guidelines

1. You are expected to submit **a single pdf file** per group (and **not** per person). **Avoid** compressed folders. If you upload something which is not a single pdf file, I'll ask you to resubmit.

2. The **name of the pdf file must be** "`NumOptReport2425_`" followed by the family names of the team members, in alphabetical order; for example, for a team of students with family names *Rossi*, *Bianchi*, and *Verdi*, the pdf file must be

   `NumOptReport2425_Bianchi_Rossi_Verdi.pdf`

3. The file should be submitted through the "**Consegna elaborati**" tab on the course page.

4. 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.

5. The **deadline** for submission is **one week before the date of the official call** at which you aim to take the exam.

6. **People in the same group are expected to take the exam in the same call**. If a team member fails the exam or rejects the mark he/she can take the exam in another call maintaining the same report.

5 Discussion guidelines

1. For the discussion of the report, **the team must bring a PC** with all the codes ready to be run, in case of needs, and the pdf of the uploaded report.

2. The team will have **12 minutes to present the results** contained in the report. After the presentation, there will be a discussion with the teacher, asking questions to the team members.

3. The team members will discuss the report directly commenting on it. **No slides, nor similar kind of presentations**.

Bibliography

[1] https://www.researchgate.net/publication/325314497_Test_Problems_for_Unconstrained_Optimization
[2] https://www.researchgate.net/publication/45932888_Test_Problems_in_Optimization