Exercises

# Continuous Optimization

Prof. Dr. Peter Ochs

www.mop.uni-saarland.de/teaching/OPT24

— Summer Term 2024 —

## — Assignment 2 —

### Exercise 1. [10 points]

The goal of this exercise is to come up with an algorithm to solve a system of linear equations $Ax = b$ using optimization techniques.

Consider a quadratic function $\varphi(x) = \frac{1}{2}x^\top Ax - b^\top x$, where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Then perform the following tasks.

(i) Show that finding the minimizer of $\varphi(x)$ is equivalent to finding solution to the system $Ax = b$.

(ii) Let $r^{(k)} = b - Ax^{(k)}$, $(k = 1, 2\ldots)$ be the residual (error) at the $k^{th}$ step. Show that $r^{(k)}$ is the direction of steepest descent for the function $\varphi(x)$.

(iii) Following the direction of steepest descent (check Algorithm 7.1 in the lecture notes), the iterations are given by $x^{(k+1)} = x^{(k)} + \tau_k r^{(k)}$ where $\tau_k > 0$ is the step length. Find an exact expression for the step length parameter $\tau_k$ by solving the optimization problem 7.6 from lecture notes. That is, solve

$$\min_{\tau_k > 0} \varphi(x^{(k)} + \tau_k r^{(k)})$$

and derive an exact expression for $\tau_k$.

### Exercise 2. [5 points]

Consider the function $f(x) = x^2$ and let, for every $k \in \mathbb{N}$,

$$d^{(k)} = -1,$$
$$\tau_k = 2^{-k-1}.$$

(i) Show that the given $d^{(k)}$ is a descent direction for every $x^{(k)} > 0$.

(ii) For $x^0 = 2$, show using induction, that the descent method gives iterates of the form $x^{(k)} = 1 + 2^{-k}$. (*Show all the induction steps*).

(iii) Do these iterates converge to the minimizer $x = 0$ ?

(iv) Comment on what went wrong and how could it be possibly related to the Wolfe's conditions ?

### Exercise 3. [5 points]

Consider the so called **2D-Rosenbrock function**

$$f(x_1, x_2) := (a - x_1)^2 + b(x_2 - x_1^2)^2 ,$$

where $a \in \mathbb{R}$ and $b > 0$ are parameters.

(a) Show that $f$ has a global minimizer at $x = (a, a^2)$.

(b) Show that $\nabla f$ is not Lipschitz continuous.

(c) Show that the function $f$ is not convex.

### Exercise 4. [20 points]

*Programming part for the problem in Exercise 3.* The goal of this programming exercise is to find a suitable approximation to a global minimizer of the 2D-Rosenbrock function using Gradient Descent based algorithms. You have been provided with three files `ex02_rosenbrock.py`, `GradientDescent.py`, `InExactLineSearch_GD.py`.

- `ex02_rosenbrock.py` generates the problem data, sets the initial point, the stopping criterion (tolerance value for the residual), the maximal number of iterations, calls the Gradient Descent Algorithm with constant step size, the Gradient Descent Algorithm with Backtracking Line Search and the Gradient Descent Algorithm with an Inexact Line Search Method, collects the results, and generates two plots. The first plot shows "number of iterations vs. residual" (or "time vs. residual"; see code) and the second plot draws the level lines of the objective and the iterates of the algorithms that were used.

Implement the missing code; see `TODO`s. Explanations are given in the file.

- In `GradientDescent.py`:
  - Initialization (2 points).
  - Compute the gradient (2 points).
  - Implement the Backtracking Line Search (6 points).
- In `InExactLineSearch_GD.py`:
  - Implement the inexact line search, where in each iteration we solve the following problem approximately using a numerical solver

    $$\min_{\tau>0} f(x^{(k)} - \tau \nabla f(x^{(k)}))\,, \qquad (1)$$

    where $x^{(k)}$ is the iterate at $k^{\text{th}}$ iteration. Please use `scipy.optimize.minimize`[1] with `L-BFGS-B`[2] as the method parameter to solve the minimization problem in (1). (5 points).
- In `ex02_rosenbrock.py`:
  - Find parameters such that all algorithms find the same solution (residual smaller than `tol=1e-2`) within the given number of `maxiter=7500` iterations, i.e. `breakvalue=2` should be returned from all functions. (5 points)

**Submission Instructions:** This assignment sheet comprises the theoretical and programming parts.

- **Theoretical Part:** Write down your solutions clearly on a paper, scan them and convert them into a file named *theory.pdf*. Take good care of the ordering of the pages in this file. You are also welcome to submit the solutions prepared with LaTeX or some digital handwriting tool. You must write the full names of all the students in your group on the top of first page.

[1] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`
[2] `https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html#optimize-minimize-lbfgsb`

- **Programming part:** This assignment sheet also contains a programming exercise in the form of `ex02_rosenbrock.py`, `InExactLineSearch_GD.py`, `GradientDescent.py` file. Submit your solution for the programming exercise with the filename `ex02_rosenbrock_solution.py`, `InExactLineSearch_GD_solution.py`, `GradientDescent_solution.py`. You can only use python3.

- **Submission Folder:** Create a folder with the name *MatA_MatB_MatC* where MatA, MatB and MatC are the matriculation number (Matrikelnummer) of all the students in your group; depending on the number of people in the group. For example, if there are three students in a group with matriculation numbers 123456, 789012 and 345678 respectively, then the folder should be named: *123456_789012_345678*.

- **Submission:** Add the files, namely, *theory.pdf*, `ex02_rosenbrock_solution.py`, `InExactLineSearch_GD_solution.py`, `GradientDescent_solution.py` to your submission folder and compress the folder into *123456_789012_345678.zip* file and upload it on the link provided on Moodle.

- **Deadline:** The submission deadline is 07.05.2023, 2:00 p.m. (always Tuesday 2 p.m. ) via Moodle.