Assignment III

MeMAD

August 30, 2025

Abstract

The goal is to revise some relevant aspects of the line search optimization algorithms, especially the steepest descent method. Please upload your solutions in one compressed file to Classroom before September 9th.

Problem 1

Suppose that $f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x}$ where $Q \in \mathbb{R}^{n\times n}$, $Q \geq 0$, $Q = Q^T$. Show that $f(\mathbf{x})$ is convex $\forall \mathbf{x} \in \mathbb{R}^n$.

Problem 2

Let it be $A \in \mathbb{R}^{n\times n}$, $A = A^T$. Consider

$$B = A + \alpha\mathbb{I},$$

where $\mathbb{I} \in \mathbb{R}^{n\times n}$ denotes the identity matrix and $\alpha \in \mathbb{R}^+$. Show that $B > 0$ for "big enough" values of α.

Problem 3

Make a script in Python to generate a symmetric random matrix $A \in (-0.5, 0.5)^{10\times 10}$. Use the idea of problem 2 to find a suitable α to build B such that $B > 0$. Explain why the "inner structure" of both A and B remain the same by inspecting their spectrum.

Problem 4

Explain why the idea of problem 2 fails if $A \neq A^T$.

Important

For problems 5 - 8 consider the following functions $f : \mathbb{R}^n \to \mathbb{R}$:

- Translated sphere's function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (x_i - c_i)^2, \quad \text{for a given (fixed) } \mathbf{c} \in \mathbb{R}^n. \tag{1}$$

  You can let it be $\mathbf{c} = (1, 1, 1 \cdots, 1)$ for instance.

- Rosenbrock's function:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \Big[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \Big]. \tag{2}$$

- Perm's function n, β:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} (j^i + \beta) \left(\left(\frac{x_j}{j} \right)^i - 1 \right) \right)^2, \quad \text{for a given (fixed) } \beta \in \mathbb{R}. \tag{3}$$

  You can let β be 1 for instance.

Also, as the started point consider $\mathbf{x}_0 = (0.5, 0.5, \cdots, 0.5)$. Furthermore, you can assume $n = 5$.

Problem 5

Compute analytically $\nabla f(\mathbf{x})$ for functions (1)-(3).

Problem 6

Make a Python script implementing the Steepest Descent (SD) algorithm for functions (1)-(3). Use a fixed length step and the analytical gradient. Show plots of the iterations versus the function value.

Problem 7

Improve your previous script by using a numerical gradient instead of the analytical one. Next, solve the minimization problem for functions (1)-(3) using the same number of iterations you did as before. Make a comparison of the solutions you get with those of problem 6.

Problem 8

Improve further your script by adding linear decreasing, adaptive and intelligent (i.e. Armijo or Wolfe conditions) length steps. You can use either an analytical or numerical gradient. Use some plots to make a comparison of the results you get for each of the four different length steps.

Problem 9

Consider the following function

$$f(x_1, x_2) = 10^9 x_1^2 + x_2^2. \tag{4}$$

Consider $\mathbf{x}_0 = (1.5, 1.5)$ as starting point. Solve the minimization problem using your SD script. How many iterations does your SD implementation need to achieve a function value less than $1e^{-4}$? Next, scale the variables of (4). How many iterations does your SD implementation need to achive a funcion value lower than $1e^{-4}$ in this scaled version of (4)?

Problem 10

Analytical and numerical differentiation aren't the only ways to compute derivatives. Do a little research about automatic differentiation. Then, consider the Himmelblau function:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2. \tag{5}$$

Compute the partial derivatives of (5) at the point $(1, -1)$ using automatic differentiation (draw on paper the corresponding graphs and show your procedure). What advantages and disadvantages do analytical, numerical and automatic differentiation have with respect to each other?