Nonlinear Optimization I, Fall 2021
Homework 1
Due before lecture on 9/14

Your submitted solutions to homeworks should be entirely your own work. Do not copy solutions from other students or any online source. You are allowed to discuss homework problems at a high-level with other students, but should carry out the execution of any thoughts/directions discussed independently, on your own. Feel free to cite any result presented in class without proof.

You can write solutions by hand or type them up (the LaTeX code for this pdf is on blackboard).

Q1. Compute $\nabla f(x)$ and $\nabla^2 f(x)$ for the following functions $f\colon \mathbb{R}^n \to \mathbb{R}$

(a) $f(x) = \frac{1}{2}x^T Hx$ where $H \in \mathbb{R}^{n\times n}$ is a fixed matrix. What if H is symmetric?

(b) $f(x) = b^T Ax - \frac{1}{2}x^T A^T Ax$, where $A \in \mathbb{R}^{m\times n}$ is a fixed matrix and $b \in \mathbb{R}^m$ is a fixed vector.

(c) $f(x) = \|x\|_2 = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$.

(d) $f(x) = \|Ax - b\|_2$, where $A \in \mathbb{R}^{m\times n}$ is a fixed matrix and $b \in \mathbb{R}^m$ is a fixed vector.

Q2. Consider any set $S \subseteq \mathbb{R}^n$ and twice continuously differentiable function $f\colon S \to \mathbb{R}$. Let $x \in S$ and $s \in \mathbb{R}^n$ be such that $x + ts \in S$ for all $t \in [0, 1]$.

(a) By defining $\theta(t) = f(x + ts)$ and using the Fundamental Theorem of Calculus:

$$\theta(1) = \theta(0) + \int_0^1 \theta'(t)\ dt,$$

show that

$$\left|f(x+s) - f(x) - \nabla f(x)^T s\right| \leq \frac{1}{2}L\|s\|_2^2$$

whenever f has an L-Lipschitz continuous gradient on S.

(b) Justify the formula

$$\theta(1) = \theta(0) + \theta'(0) + \int_0^1 \int_0^t \theta''(\alpha)\ d\alpha\ dt.$$

Hence, show that

$$\left|f(x+s) - f(x) - \nabla f(x)^T s - \frac{1}{2}s^T \nabla^2 f(x)x\right| \leq \frac{1}{6}Q\|s\|_2^3$$

whenever f has an Q-Lipschitz continuous Hessian on S.

Q3. : Let x^* be a global minimizer of a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$. Prove that if $\nabla^2 f(x^*) \succ 0$ then x^* is, in fact, the unique global minimizer of f.

Q4. Please read the syllabus defining the linear optimization problem that will be solved to maximize each student's course score. This question asks you to reason about and then write a short program to solve this three-dimensional optimization problem.

(a) Denote the set of all feasible grading rubrics $(H, M, F) \in \mathbb{R}^3$ as

$$\mathcal{P} = \{(H, M, F) \mid H+M+F \leq 100,\ H, M \geq 15,\ F \geq M,\ 50 \leq M+F \leq 80,\ H+M+F \geq 90\}.$$

Prove the set $\mathcal{P}$ is convex (that is, for every $x, y \in \mathcal{P}$ and $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in \mathcal{P}$).

(b) Denote the set of ten corners of $\mathcal{P}$ as

$$\mathcal{S} = \{(15, 40, 40),\ (20, 40, 40),\ (50, 25, 25),\ (40, 25, 25),\ (15, 37.5, 37.5),\\ (15, 15, 65),\ (20, 15, 65),\ (50, 15, 35),\ (40, 15, 35),\ (15, 15, 60)\}.$$

Carathéodory's Theorem ensures us that $x \in \mathcal{P}$ can be written as a convex combination of these corners: For any $x \in \mathcal{P}$, there exist coefficients $\lambda_p \geq 0$ for each $p \in \mathcal{S}$ such that

$$\sum_{p \in S} \lambda_p p = x \qquad \text{and} \qquad \sum_{p \in S} \lambda_p = 1.$$

Given this theorem, show that for any student with arbitrary course component grades (C_H, C_M, C_F, C_P), one of these ten corner points in $\mathcal{S}$ maximizes their course score.

(c) Knowing some corner must be optimal, write a program that computes the maximum course score for students given their four component gradings. Compute and output the (i) maximum course score and (ii) an optimal corner point in $\mathcal{S}$ for the following three hypothetical students

$$(C_H, C_M, C_F, C_P) = (100, 90, 80, 70),$$

$$(C_H, C_M, C_F, C_P) = (85, 85, 85, 85),$$

$$(C_H, C_M, C_F, C_P) = (70, 80, 90, 100).$$

General Guidelines for Programming HW Problems: You can do programming assignments in any programming language you feel comfortable with (python, matlab, java, c/c++, haskell, etc). Programming questions will ask for you to solve a particular problem or describe particular settings to run an algorithm under. You must submit both your code and the requested output/plots from running your code. Grading will focus primarily on the quality of these outputs rather than of your code.