PURDUE UNIVERSITY · CS 52000
COMPUTATIONAL OPTIMIZATION

HOMEWORK
David F. Gleich
January 26, 2026

Please answer the following questions in complete sentences in a clearly prepared manuscript and submit the solution by the due date on Gradescope.

Remember that this is a graduate class. There may be elements of the problem statements that require you to fill in appropriate assumptions. You are also responsible for determining what evidence to include. An answer alone is rarely sufficient, but neither is an overly verbose description required. Use your judgement to focus your discussion on the most interesting pieces. The answer to "should I include 'something' in my solution?" will almost always be: Yes, if you think it helps support your answer.

Problem 0: Homework checklist

- Please identify anyone, whether or not they are in the class, with whom you discussed your homework. This problem is worth 1 point, but on a multiplicative scale.
- Make sure you have included your source-code and prepared your solution according to the most recent Piazza note on homework submissions.

Problem 1: Optimization software

We'll be frequently using software to optimize functions, this question will help familiarize you with a piece of optimization software relevant for our study.

The function we'll study is the square root Rosenbrock function:

$$f(x) = \sqrt{100(x_2 - x_1^2)^2 + (1 - x_1)^2)}.$$

1. Show a contour plot of this function
2. Write the gradient and Hessian of this function.
3. Check that your gradient and Hessian of this function are correct by comparing against an automatic differentiation tool like those used in `Flux.jl` as we did in class or something similar in Python.
4. By inspection, what is the minimizer of this function? (Feel free to find the answer by other means, e.g. looking it up, but make sure you explain *why* you know that answer *must* be a *global* minimizer.)
5. Explain how any optimization package could tell that your solution is a local minimizer.
6. Use Optim.jl (or Poblano for Matlab, or Scikit.learn for python) to optimize this function starting from a few different points. Be adversarial if you wish. Does it always get the answer correct? Show your code. You must use a call/tool where you provide the gradient.

Problem 2: Optimization software

Repeat all the steps for problem 1 for the Himmelblau function

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

Using the same algorithm as in problem 1, which function takes the most iterations when starting from the point (3,3).

Problem 3: Optimization theory

Suppose that $f : \mathbb{R} \to \mathbb{R}$ (i.e. is univariate) and is four times continuously differentiable.

1. Show that the following conditions imply that x^* is a local minimizer.

   i. $f'(x^*) = 0$

   ii. $f''(x^*) = 0$

   iii. $f'''(x^*) = 0$

   iv. $f''''(x^*) > 0$

2. Ask an LLM to give you a univariate function with a local minimizer at $x = 2$ where those conditions are false.

Problem 4: Convexity

Convex functions were all the rage when I started teaching this class, and one of the interests of students in this class. Let's do some matrix analysis to show that a function is convex. Solve problem 2.7 in the textbook, which is:

> Suppose that $f(x) = x^T Q x$, where Q is an $n \times n$ symmetric positive semi-definite matrix. Show that this function is convex using the definition of convexity, which can be equivalently reformulated:
>
> $$f(y + \alpha(x - y)) - \alpha f(x) - (1 - \alpha) f(y) \leq 0$$
>
> for all $0 \leq \alpha \leq 1$ and all $x, y \in \mathbb{R}^n$.

This type of function will frequently arise in our subsequent studies, so it's an important one to understand.

Optional Problem 5: Fool an LLM

Can you find a function f where the LLM gives you the wrong gradient when you ask it to solve a problem like 1 or 2?