Linear and Nonlinear Optimization, Fall 2022
MATH-UA 253 and MA-UY 3204

Lectures: Tu/Th 3:30-4:45PM CIWW 101.
Recitations: Fri 3:30-4:45PM CIWW 101.

Updated October 13, 2022.

Instructor:	Sam Potter	Email: `sfp@cims.nyu.edu`
		Web: `sampotter.github.io`
		Office hours: *TBD.*
Teaching assistant:	Mariana Martinez-Aguilar	Email: `mgm9745@nyu.edu`
		Office hours: *TBD.*

1 Overview

1.1 Description

Here's the description of the course, as per `https://math.nyu.edu/dynamic/courses/undergrad/math-ua-253/`:

> *Optimization is a major part of the toolbox of the applied mathematician, and more broadly of researchers in quantitative sciences including economics, data science, machine learning, and quantitative social sciences. This course provides an application-oriented introduction to linear programming and convex optimization, with a balanced combination of theory, algorithms, and numerical implementation. While no prior experience in programming is expected, the required coursework will include numerical implementations, including some programming; students will be introduced to appropriate computational tools, with which they will gain experience as they do the numerical assignments. Theoretical topics will include linear programming, convexity, duality, minimax theorems, and dynamic programming. Algorithmic topics will include the simplex method for linear programming, selected techniques for smooth multidimensional optimization (eg Newton's method and the conjugate gradient method), techniques for solving for L1-type optimizations, and stochastic gradient descent. Applications will be drawn from many areas, but will emphasize economics (eg two-person zero-sum games, matching and assignment problems, optimal resource allocation), data science (eg regression, convex-relaxation-based approaches to sparse*

inverse problems, tuning of neural networks, prediction with expert advice) and operations research (eg shortest paths in networks and optimization of network flows).

We will follow this closely, but may add or remove topics, depending on how things go.

1.2 Prerequisites

Again, per `https://math.nyu.edu/dynamic/courses/undergrad/math-ua-253/`:

MATH-UA 123 Calculus III OR MATH-UA 129 Honors Calculus III OR MATH-UA 213 Math for Economics III (for Economics majors) with a grade of C or better and/or the equivalent, AND MATH-UA 140 Linear Algebra OR MATH-UA 148 Honors Linear Algebra with a grade of C or better and/or the equivalent.

Please understand the following: we will freely use the concepts that are likely to have been covered in these courses. It's perfectly fine for you to have forgotten some of this material, but please be aware that there isn't time to review this material in detail during class. It is likely that there will be material that you need to review from these classes as it comes up during class. This is a fact of life when it comes to studying math. Please view this course as an opportunity to *develop expertise* with the material from the prerequisite courses. For example, learning the multivariable Newton's method for minimization will give you a much deeper understanding of the Hessian. Developing this kind of expertise takes time.

About programming: This course will require you to do some programming. You are not expected to have programmed before, but the reality is that this course will prove challenging for you if you have absolutely no programming experience. Please bear this in mind. If you have the time and inclination, this course is a good opportunity to learn how to program. We will exclusively use Python in this course.

2 Lectures notes and textbooks

Officially, no textbook is required for the course. Instead, lectures notes will be made available online as they are written. We will primarily follow:

- "Linear and nonlinear optimization" by Griva, Nash, and Sofer.

In addition, we will likely dip into the following textbooks:

- "Linear Programming: Foundations and Extensions" by Vanderbei, and
- "Linear and Nonlinear Programming" by Luenberger and Ye.

These are both available online through the library using your NYU account.

There is a good chance we will draw examples and material from the following textbooks:

- "Linear Algebra and Learning From Data", Strang.
- "Numerical Optimization", Nocedal and Wright.
- "Convex Optimization", Boyd. (Easy to find online.)

It will be made clear—both in the course schedule and in the lecture notes—which sections from these references are being drawn from.

3 Computing

We will use the Python programming language for homework assignments. Mariana will run a computing lab during recitation, where you will solve small problems related to what we're doing in class in order for you to learn how to use Python. We will make extensive use of NumPy, SciPy, and Matplotlib. During the first recitation, you will learn how to install Python and get set up.

4 Course requirements and grading

Finding a buddy. There will be a "Homework 0" where each person needs to find a "buddy". You must find one other person in class to exchange contact information with—and then give me some proof that you have done so. Homework 0 will count for one homework assignment worth of extra credit.

Homework. There will be ~10 homework assignments throughout the course. They will feature problems with a mix of writing and programming. For each assignment, you will be expected to submit a PDF written using LaTeX. You will usually be asked to make some plots or tables of numbers, which must be included inline in your LaTeX submission. Programming assignments will be submitted as Python scripts which can be run by the grader to generate the solutions to the homework problems.

You can use Overleaf to prepare your homework submissions, or you can do it locally—either way, the tutorial material provided by Overleaf is excellent. If you want to learn how to do something, I recommend looking around on Overleaf's website first. Here is a 30 minute introduction to LaTeX.

Midterm and final. There will be an in-class midterm and in-class final. You will be allowed a single page of notes, both sides. It will last the whole class period. In it, you can be expected to answer a mix of problems, for example:

- *Tell me about something we learned in class.* This will be fairly open-ended. For example: *"What is Newton's method for minimization? Tell me three significant facts about it that we learned."*
- *Calculate something by hand* (e.g., apply an algorithm to a simple problem, or some other "calculation"—i.e., not a "proof").
- *Derive something simple.* I will *not* ask you to prove (what I consider to be) a challenging theorem on a test. This may be something you haven't seen before. In this case, I will try to make the question as straightforward as possible—I may include multiple parts to guide you along.
- *Sketch a theorem we saw in class.* I may ask you to give me a sketch of a proof of some result we saw in class. You will not need to reproduce the proof in exact detail, but you will need to explain (using some math!) the general outline. This will require you to understand how the proof works, rather than committing it to memory (which is a waste of time).
- *Answer a question about the programming homework.* I may ask you to tell me something about a recent programming homework. For example: "*If we used Algorithm X instead of Algorithm Y to solve Problem Z, what would happen?*" I *will not* ask you to "write code" on an exam (again, a waste of time).

My goal is for these tests to feel as fair as possible, and to incentivize you to do all the homework, come to every class, and come to every recitation. This is the only way that you will get the most out of the class. If you take full advantage of the class and study, the tests should feel straightforward.

Recitations. Recitations will primarily be a "computing lab", led by the TA. In these labs, you will learn to solve problems similar to what is being covered in class, and what will be required on the homework. The labs will exclusively use Python.

There will be a small part of your grade based on recitation attendance. You will not need to do any assignments, but attending recitation is a key part of the course. You will not get any "free absences", but the grade will be small enough that missing a few will not adversely impact your grade—it is not necessary to get 100% in this class to get an A.

5 Course requirements and grading

The weight of each assignment is as follows:

Find a buddy	EC (1HW)
Recitation	10%
Quizzes	10%
Homework	25%
Midterm	20%
Final exam	35%