LINEAR AND NONLINEAR OPTIMIZATION, FALL 2021
MATH-UA 253 and MA-UY 3204
Lectures Tues/Thurs 3:30-4:45 (WWH 109), Recitations Fri 3:30-4:45 (Silver 411)

updated September 7, 2021[1]

Instructor:	Miranda Holmes-Cerfon (she/her)	E-mail: holmes@cims.nyu.edu Tel: 212.998.3262 Office: WWH 1111 Office Hours: *TBD*
Teaching Assistant:	Shanyin Tong (she/her)	Email: shanyin.tong@nyu.edu Office: WWH 1106 Office Hours: *TBD (probably online)*

This syllabus is a work in progress. Items that may be revised are written in this color.

Description: 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 *nonlinear optimization*, with a balanced combination of theory, algorithms, and numerical implementation. Theoretical topics will include linear programming, convexity, duality, Lagrange multipliers, and minimax theorems. Algorithmic topics will include the simplex method for linear programming, and selected techniques for smooth multidimensional optimization, such as steepest descent, accelerated steepest descent stochastic gradient descent, Newton's method, and the use of barriers or penalties for constrained optimization. Applications will be drawn from many areas, but will emphasize economics (eg two-person zero-sum games, matching and assignment problems), data science (eg regression, convex-relaxation-based approaches to sparse inverse problems, tuning of neural networks) and operations research (eg shortest paths in networks and optimization of network flows). The coursework will require numerical implementations, including some programming. No prior experience in programming is expected; students will be introduced to appropriate computational tools as they work through numerical assignments.

The fundamental problems considered by linear programming, and nonlinear optimization, are described at the end of this syllabus.

Prerequisites: multivariable calculus and linear algebra. The multivariable calculus prerequisite is satisfied by MATH-UA 123 (Calculus III) or MATH-UA 129 (Honors Calculus III) or MATH-UA 213 (Math for Economics III) with a grade of C or better. The linear algebra prerequisite is satisfied by MATH-UA 140 (Linear Algebra) or MATH-UA 148 (Honors Linear Algebra) with a grade of C or better. The prerequisites can also be met by equivalent coursework elsewhere.

[1] *Main changes from the previous syllabus: added a new textbook, Nocedal&Wright; from previous: added a section at the end, describing the fundamental problems in linear programming and nonlinear optimization.*

Textbooks: We will use the following two books as texts. Each is available to members of the NYU community as a free pdf download from SpringerLink, and a paperback copy can be purchased at a very low cost from the book's SpringerLink page; for access, find the book in Bobcat (or use the permalink given below) then click on SpringerLink.

(1) Robert Vanderbei, Linear Programming – Foundations and Extensions, 5th edition, Springer, 2020

(2) David Luenberger and Yinyu Ye, Linear and Nonlinear Programming, 4th edition, Springer-Verlag, 2015

We will cover only selected parts of each book. For a few topics that aren't covered well by these books, additional notes or readings will be made available.

Additional books: This book, while more suitable as a graduate textbook, has an excellent presentation of both linear and nonlinear optimization, and we will assign optional readings from it. It is also available from SpringerLink.

(3) Jorge Nocedal and Stephen J. Wright, Numerical Optimization, 2nd edition, Springer Science & Business Media, 2006

The following books, while not suitable for use as textbooks for this class, are rich with applications to economics and data science:

(a) Rakesh Vohra, Advanced Mathematical Economics, Routledge, 2005

    This book is downloadable for free via Bobcat (use the permalink then choose online access). Its scope is broader than just optimization, and its level is more advanced than this class. But the first 115 pages are pretty accessible, and the topics they cover are within the scope of this class. The book is full of applications to economics – far more than we'll have time for.

(b) Gilbert Strang, Linear Algebra and Learning from Data, Wellesley Cambridge Press, 2019. Like Vohra, this book's level is more advanced than our class; moreover its focus isn't mainly on optimization (in fact, the emphasis is on tools from linear algebra). But the book is nevertheless worth mentioning, since it offers a broad perspective on how optimization, linear algebra, and related tools are used in data science. Unfortunately, it isn't available electronically, and only NYU Shanghai has a physical copy (on reserve). The book's website offers links to several sellers, and Professor Strang's website `http://www-math.mit.edu/~gs/` has links to youtube and MIT OCW sites where you'll find lectures from a 2018 course based on a draft of the book.

Computing: The required coursework will include numerical implementations, including some programming. Students will be introduced to the open-source Julia programming language and JuMP optimization package early in the semester. Julia is rather similar to MATLAB and Python, so it will be advantageous if you are familiar with one of these languages. Note that using the Julia software package and basic programming will be an essential and obligatory part of this course.

Information about downloading and installing Julia and JuMP will be distributed before the first recitation. The first two recitations will be devoted to Julia/JuMP tutorials, to help you get started using these computational tools. *Students are strongly urged to attend the Julia/JuMP tutorials in person, to make sure they have access to our computational tools.*

Course requirements and grading: The course requirements are

- **Exams:** There will be 1 midterm exam and 1 final exam, scheduled for

  Midterm: **Thursday, October 28, 3:30-4:45**, in class.

  Final exam: **Thursday, December 16, 4pm-5:50pm**. Location will be set by the registrar.

- **Homework:** There will be homework assignments, roughly one every two weeks. Each assignment will count equally toward your grade (regardless of the max possible points.) Homework should be submitted on Gradescope, which you can access via Brightspace. Late homework will be deducted 10% per day.

- **Quizzes:** There will be weekly quizzes. They will be short and you'll be able to do them quickly, provided you are up to date on the material. The quizzes will help make sure you don't forget about this class during the periods when no lectures are scheduled; in particular, they'll help you remember to stay up to date. They will usually be posted to Gradescope by Friday at 12pm with a completion deadline just before the following week's first lecture (which is usually on Tuesday). You may use your notes and textbook in order to complete them, however you are encouraged to first attempt the questions without extra materials, to test your current knowledge. Each quiz will count equally.

- **Numerical Assignments / Participation:** *The course will contain a participation component whose nature is still to be determined. This may take the form of numerically-oriented "lab" sessions & worksheets, which will be graded mostly for participation. These will likely take place during recitations, where you will work in groups on the computational problems. More instructions about the nature of these assignments and how to submit responses will be provided later.*

- **Recitations:** You are each registered for a recitation, and are expected to attend each week (provided you are not sick – see end of syllabus.) In the recitation you will work on problems, both theoretical and on a computer. You will also sometimes work on Numerical Assignments, in teams, completion of which will be mandatory. *Bringing a laptop to recitation when there are computational problems is strongly encouraged.*

Grades will be calculated using the following weights:

Final Exam	35%
Midterm	20%
Quizzes	10%
Homework	25%
Numerical Assignments / Participation	10%

Overall goals and a tentative semester plan: Our material has two related but distinct threads, namely nonlinear optimization (for which we'll mainly use the book of Luenberger & Ye), and linear programming (for which we'll mainly use Vanderbei's book.) We'll focus on unconstrained nonlinear optimization in the first part of the course, and linear programming in the second part. In the last few classes we will discuss constrained nonlinear optimization.

The class will emphasize connections with and applications to economics and data science. These are sometimes conceptual (such as the use of linear programming to solve two-person zero-sum games) and sometimes quite practical (such as the use of stochastic gradient descent to tune neural networks – in effect, minimizing a nonlinear and nonconvex function of many variables). As we discuss various optimization techniques, we will focus on the technique's essential character, power, and limitations; computing with Julia and JuMP will permit us to do examples, bringing the methods to life and applying them.

Tentatively, we expect to cover the following topics (in approximately this order):

- Introduction to linear programming and nonlinear optimization: some example problems.
- Nonlinear optimization; gradient descent, accelerated (momentum) gradient descent, stochastic gradient descent (Luenberger & Ye, chapter 8)
- Newton's method, and quasi-Newton methods (drawing from Luenberger & Ye 10.1-10.4)
- Introduction to linear programming; geometry of the simplex method (Vanderbei, chapters 1-3)
- Duality in linear programming (Vanderbei, 5.1-5.5) and its application to game theory (Vanderbei, chapter 11).
- Some applications of linear programming to data science (Vanderbei, chapter 12).
- Linear programming applied to network problems, and to matching and assignment problems (Vanderbei chapters 14 & 15, with additional sources for matching and assignment problems)
- Constrained nonlinear optimization: barrier & penalty methods for nonlinear optimization with constraints (drawing from Luenberger & Ye chapters 5 and 13)

Some Practical Matters

Gradescope We will use Gradescope for homework and quizzes. You'll find Gradescope in the "Content / Gradescope" section of the Brightspace site. We will inform you each time we post something on Gradescope.

Class materials All other class materials (e.g. pdf versions of lecture notes, extra handouts, homework and quiz solutions) will be available in the "Content / Course Materials" section of the Brightspace site.

Readings Readings from the textbook for the next week's lectures will be posted to "Content / Weekly Readings" section of the Brightspace site. Readings for the following week will usually be posted before Friday at 12pm. You are strongly encouraged to do the readings before coming to class on Tuesday.

Class communication forum We will use the "Discussions" forum in Brightspace. If you have questions, find a typo in an assignment, need help with a problem, have a question about logistics, etc, please ask there. If you know an answer to another student's question, please help them; we will try to keep an eye on the forum but would like to encourage you to also interact and help with each other. You may also use this forum to find other students with whom to study – it is always a good idea to study with other students; even more so if you must miss classes because of covid-rules.

If you have a question that concerns you individually, please email the instructor. I will do my best to respond within 24 hours on weekdays (I will not necessarily respond on a weekend.) If your question could be of interest to others in the class, please post it in the forum instead – if you email the instructor with such a question I will ask that you post in the forum so the instructor & TA can respond there, and then others can also benefit from the response.

Office Hours All students are welcome to attend the instructor's and TA's office hours. You may come with specific questions about the course material, or you may come simply to hear other students' questions and to work on problems with them. If you would like to discuss an issue privately, please email the instructor ahead of time so we can set aside time to meet in private (if this is during office hours we will respectfully ask other students to leave during that time.)

Quirks of the NYU Spring 2021 calendar

Thursday Sept 2:	first day of class
Friday Sept 10:	first day of recitation
Tuesday Oct 12:	no class (scheduled as a Monday in NYU's calendar)
Thursday November 25:	no class (Thanksgiving)
Friday November 26:	no recitation (Thanksgiving)
Tuesday December 14:	last day of class (this will be a review class since it is only 2 days before the final exam.)

Some Policies

Collaboration on homework Students are welcome – and even encouraged – to discuss the homework problems with others. However, for both numerical and pencil/paper type questions, each student must implement and present their own solutions (this is an important part of the learning process). Direct copying of another student's solution is *not* permitted – both because it amounts to cheating, and because it defeats the entire purpose of the homework (which is to gain familiarity with new concepts and techniques).

Academic integrity Plagiarism and cheating will not be tolerated. NYU's College of Arts and Sciences has policies in this area, and they will be followed. See http://cas.nyu.edu/academic-integrity.html

Class participation The class time will be a mixture of lectures, and exercises that students will be expected to work on, individually or in teams. If you attend class, the expectation is that you will fully participate – you will work on problems, share your thoughts with other students or with the class, etc. I may call on students individually to answer problems – you are encouraged to attempt to answer; even if you aren't sure of the answer you will learn from trying – but you are always welcome to pass. Note that I have class rosters with students' legal names. I will gladly honor your request to address you by an alternate name or pronounciation or gender pronoun; please advise me accordingly early in the semester so that I may make appropriate changes to my records.

Respect Simultaneously, you are expected to show respect to all students – regardless of their their interpretation of the class material, their identity along any axis, etc. Every student should feel welcome to say anything (respectful) at any moment, and should not be afraid of making mistakes. Each student learns in a different way, and interprets and understands mathematics in a different way. My philosophy is that we learn best when we work directly on problems, when we get feedback from others, and when we make mistakes – so I intend class time to be a place where you can feel free to test ideas, to possibly make mistakes, to push yourself intellectually into places where you are uncomfortable, and then to learn from your own and from other students' intellectual reactions.

Covid

Covid-19 may cause some disruption to the way class would normally run. Here is a summary of some of the issues and contingency plans you may need to be aware of.

- *Please wear your mask properly, at all times.* We don't want to have another shutdown.
- If you are sick, you are supposed to stay home. Be reasonable about this – don't come to class if you have covid symptoms and haven't been tested.

- If you miss class: The instructor will post handwritten lecture notes after each lecture. You can use these to follow the class, even if you cannot attend (actually even if you can.) Almost all other mandatory class activities can be done online (quizzes, homework, etc), the only exception being exams.

- You may wish to team up with other students so you can share resources, in case one of you misses class. You can use the Brightspace Discussion forum to find study partners; let me know if I can help facilitate this.

- Some office hours for the instructor or the TA may be moved online if a sufficient number of students are absent. *TBD – a way to let the instructor know that you are unable to attend because you are sick.*

- If the instructor or TA cannot enter the building because of covid-rules (e.g. they are sick, or caring for a sick or quarantining child), then the class will be held on zoom (provided they are healthy enough to teach.) The instructor will send a zoom link by email as early as possible (but there might not be much notice, depending on the circumstances.) Be aware therefore that the nature of the class could change at the last minute.

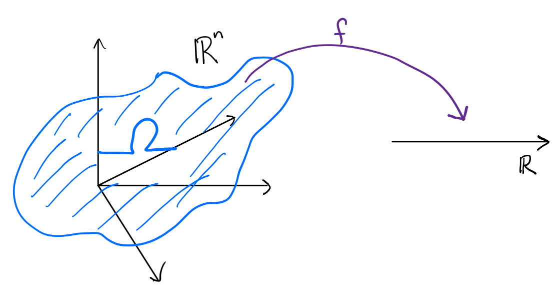
What are linear and nonlinear optimization?

The two fundamental topics we will address in this course, are nonlinear optimization, and linear optimization (usually called linear programming.) This section briefly introduces the main problem in these two areas. Of course, the richness of the topics lies in finding ways to actually solve these problems, via algorithms, to prove that these algorithms work, and to set up these problems mathematically in an application.

Nonlinear Optimization In nonlinear optimization, one starts with the following objects:

a domain $\Omega \subset \mathbb{R}^n$

a function $f : \Omega \to \mathbb{R}$.



The fundamental problem is to solve

$$\min_{x \in \Omega} f(x)\,. \tag{1}$$

In many cases, f is nonlinear (e.g. $f(x) = |x|^2$, $f(x_1, x_2, x_3) = (x_1 - x_2)^2 + e^{-x_2} x_3^5$, etc), which is why this is called nonlinear optimization. In the special case when f is linear, and Ω is defined by linear functions, the problem is called linear optimization, or linear programming (see the next section.)

We further distinguish two types of optimization problems:

- When $\Omega = \mathbb{R}^n$, the problem is called *unconstrained optimization*. In this case we optimize over all values of x.
- When $\Omega \neq \mathbb{R}^n$ (it is smaller than $\mathbb{R}^n$), the the problem is called *constrained optimization*. In this case we consider only some of the values of x; the variables are therefore constrained to a particular region in space.

Linear Programming Linear programming is a special case of (1), but has developed into its own field of study, with a rich body of theory and a variety of specialized, fast algorithms to solve the optimization problem. The fundamental problem is to solve

$$\min_{\substack{Ax=b \\ x_i \geq 0}} \sum_{i=1}^{n} c_i x_i \tag{2}$$

Here
$x \in \mathbb{R}^n$ is a vector
$b \in \mathbb{R}^n$ is a (constant) vector
$A \in \mathbb{R}^{m \times n}$ is a (constant) matrix, with $m \leq n$
$c = (c_1, c_2, \ldots, c_n)^T \in \mathbb{R}^n$ is a (constant) vector.

Problem (2) asks to minimize the linear function $c \cdot x$ over all values of x that satisfy the constraints $Ax = b$, $x \geq 0$ (we write $x \geq 0$ to mean that all components of x are nonnegative.)
There are many variations of (2) that can all be reduced to (2) by introducing extra variables (we'll learn how to do this in the course.) Some variations are

$$\min_{\substack{Ax \geq b \\ x \geq 0}} \sum_{i=1}^{n} c \cdot x\,, \quad \min_{\substack{b' \leq Ax \leq b \\ x \geq 0}} \sum_{i=1}^{n} c \cdot x\,, \quad \min_{\substack{Ax=b \\ x_1, x_2 \geq 0 \\ x_3 \ldots,\ \text{unconstrained}}} \sum_{i=1}^{n} c \cdot x\,, \quad \max_{\substack{Ax \geq b \\ x \geq 0}} \sum_{i=1}^{n} c \cdot x\,, \quad \text{(etc.)}$$

Topics in this course We'll consider, in this order:
(1) unconstrained nonlinear optimization
(2) linear programming
(3) constrained nonlinear optimization