Math 693A: Advanced Numerical Analysis
Numerical Optimization

Lecture Notes #1 — Introduction

Uduak George,
⟨ugeorge@sdsu.edu⟩

Department of Mathematics and Statistics
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

Fall 2024

Outline

1. The Class — Overview
   - Literature & Syllabus
   - Grading
   - Student Learning Outcomes
   - Expectations and Procedures
2. The Class...
   - Resources
   - Formal Prerequisites
3. Numerical Optimization
   - The What? Why? and How?
   - Concepts & Terms
   - Mathematical Formulation

Math 693A: Literature

"Required" — (An Excellent Text for Numerical Optimization)
Numerical Optimization, 2nd Edition, Jorge Nocedal and Stephen J. Wright, Springer Series in Operations Research, Springer Verlag, 2006.
ISBN-10: 0387303030; ISBN-13: 978-0387303031

"Required" — (Supplemental)
Class notes.

Math 693A: Literature

"Optional" — (A Classic in the field)
Numerical Methods for Unconstrained Optimization and Nonlinear Equations, J. E. Dennis, Jr. and Robert B. Schnabel, Classics in Applied Mathematics 16, Society for Industrial and Applied Mathematics (SIAM), 1996. ISBN 0-89871-364-1.

"Optional" — (An excellent text for Linear Algebra)
Introduction to Linear Algebra, 5th Edition, Gilbert Strang, Wellesley-Cambridge Press / Society for Industrial and Applied Mathematics, 2016. ISBN-10 0-980-23277-5, ISBN-13 978-0-980-23277-6.

Math 693A: Introduction — Grading etc.

65% `Homework:` both theoretical, and implementation (programming) — feel free to program in C/C++, Fortran, 6510 assembler, Java, Python or Matlab...

35% `Project:` Project presentation (10%) and project report (25%). Implementation of several interacting parts of an optimization package. By the end of the semester you should have a working "toolbox" of optimization algorithms which will be useful in your current and future research projects. [Complete details TBA].

Student Learning Outcomes

- Students will be able to identify objective, variables, and constraints for a given problem.
- Students will be able to apply optimization algorithms and create computer programs to solve optimization problems.

Student Learning Outcomes

- Students will be able to identify objective, variables, and constraints for a given problem.
- Students will be able to apply optimization algorithms and create computer programs to solve optimization problems.
- Students will be able to determine if an optimization algorithm applied to a model has succeeded in its task of finding a solution.
- Students will be able to understand the theoretical properties of optimization methods including convergence of the methods.

Student Learning Outcomes

- Students can recognize and give examples of contributions to numerical optimization that have been made by members of diverse cultural and gender groups and other historically marginalized people.

Expectations and Procedures, I

- Class attendance is HIGHLY RECOMMENDED —
  - Please be on time.
  - Please pay attention.
  - Please turn off mobile phones/mute your Zoom audio except when speaking to the instructor or asking a question.
  - Please be courteous to other students and the instructor.
  - Abide by university statutes, and all applicable local, state, and federal laws.
- Class attendance is **MANDATORY** for ALL in-class presentations.
- Homework and announcements will be posted on the class web page on Canvas;

Expectations and Procedures, II

- Students are expected **and encouraged** to ask questions in class!
- Please, turn in assignments on time. **Any homework that is more that 5 days late but not more than 7 days late will be worth 80% of the total point.** Any homework that is more than 7 days late will not be graded unless you make arrangement with me in advance.(The instructor reserves the right not to accept late assignments.)

Expectations and Procedures, II

- Students are expected **and encouraged** to ask questions in class!
- Please, turn in assignments on time. **Any homework that is more that 5 days late but not more than 7 days late will be worth 80% of the total point.** Any homework that is more than 7 days late will not be graded unless you make arrangement with me in advance.(The instructor reserves the right not to accept late assignments.)
- The instructor will make special arrangements for students with documented learning disabilities and will `try` to make accommodations for other unforeseen circumstances, *e.g.* illness, personal/family crises, etc. in a way that is fair to all students enrolled in the class. **Please contact the instructor EARLY regarding special circumstances.**

Expectations and Procedures, III

- Students are expected **and encouraged** to make use of office hours! If you cannot make it to the scheduled office hours: contact the instructor to schedule an appointment!

Expectations and Procedures, III

- Students are expected **and encouraged** to make use of office hours! If you cannot make it to the scheduled office hours: contact the instructor to schedule an appointment!
- Missed project presentation: Don't miss the presentation! Contact the instructor ASAP or a grade of `WU` or `F` will be assigned. A Google Link for selecting a presentation date and time is available of Canvas.

Expectations and Procedures, III

- Students are expected **and encouraged** to make use of office hours! If you cannot make it to the scheduled office hours: contact the instructor to schedule an appointment!
- Missed project presentation: Don't miss the presentation! Contact the instructor ASAP or a grade of `WU` or `F` will be assigned. A Google Link for selecting a presentation date and time is available of Canvas.
- **Academic honesty**: submit your own work — but feel free to discuss homework with other students in the class! Cite any and all sources (outside of class material) you use.

Math 693A: Computer Resources

You need access to a computing environment in which to write your code; — e.g. MATLAB, C/C++, Fortran, Python etc.

Check out `http://julialang.org/`

Free C/C++ (gcc) and Fortran (`f77`, `f95`) compilers are available for Linux/UNIX.

SDSU students can download a copy of matlab from https://library.sdsu.edu/computers-technology/software/matlab

Python (https://www.python.org/) is an open source programming language

Math 693A: Introduction — What you should know already

`MATH 524` ⇒ **Linear Algebra**

- Vector spaces, linear transformations, orthogonality, eigenvalues and eigenvectors, normal forms for complex matrices, positive definite matrices.

`MATH 340` ⇒ **Programming in Math**

- Scientific computing and computer visualization.

Math 693A: Introduction — Why???

Q: What is optimization?

A: The term **optimization** is often used to mean *improvement*, but mathematically, it is a much more precise concept: finding the **best** possible solution and often subject to some constraints.

Math 693A: Introduction — Why???

Q: Why do we need numerical optimization methods?

A: Any problem in real life where a decision needs to be made can be formulated as an optimization problem.

- Although some simple optimization problems can be solved analytically, **most practical problems** are too complex to be solved this way.

- The advent of numerical computing, together with the development of numerical optimization algorithms, has enabled us to solve problems of increasing complexity.

Math 693A: Introduction — Why???

Example of optimization problems?

- Trajectories for airplanes, space craft, robotic motion, etc.
- Shapes for cars, airfoils, aerodynamic bicycle wheels, etc.
- Risk management — investment portfolios; insurance premiums.
- Circuit and network design.

Math 693A: Introduction — Why???

`Q:` Why do we need numerical optimization methods?

`A:` Optimization problems occur in various areas, such as economics, political science, management, manufacturing, biology, physics, and engineering.

- Numerical optimization first emerged in operations research, which deals with problems such as deciding on the price of a product, setting up a distribution network, scheduling, or suggesting routes.
- Other optimization areas include optimal control and machine learning. Although we do not cover these other areas specifically in this course, many of the methods we cover are useful in those areas.

Example of an optimization problem

An optimization problem with an inequality constraint

$$\min_{x} \quad x\sin(x)$$
$$\textit{subject to} \quad 2 \leq x \leq 6.$$

- $f(x) := x\sin(x)$ is called the objective function,
- x is a variable,
- $2 \leq x \leq 6$ is an inequality constraint.

Numerical Optimization: Concepts and Term

`Students optimize:` minimize study time T, such that GPA is acceptable. `:-)`

`Nature optimizes:` Physical systems settle in a state of minimal energy —

- A ball rolls down to the bottom of a slope;
- Light rays follow the path that minimizes travel time.

In order to understand physical systems `we` must optimize: — first we must identify the **objective** (the measure of performance). The objective depends on a number of **variables** (the characteristics of the system).

Numerical Optimization: Concepts and Term

Our goal is to find the values of the **variables** that optimize (either minimize, or maximize) the **objective**. Often the variables are **constrained** (restricted) in some way (*e.g.* densities, and interest rates are non-negative).

The process of identifying the **objective**, **variables**, and **constraints** is non-trivial and will essentially be completely ignored in this class. (Usually covered in a **Mathematical Modeling** course)

Our discussion starts after the modeling is done!

Mathematical Formulation

From the point of view of a mathematician, optimization is the minimization (or maximization) of a function subject to constraints on its variables.

Notation:

$\bar{\mathbf{x}}$	*the vector of variables (a.k.a. unknowns, or parameters)*
$f(\bar{\mathbf{x}})$	*the objective function*
$\bar{\mathbf{c}}$	*the vector of constraints that the unknowns must satisfy*

The **Optimization Problem** can be written

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} f(\bar{\mathbf{x}}) \quad \textit{subject to} \quad \begin{cases} c_i(\bar{\mathbf{x}}) = 0, & i \in E \\ c_i(\bar{\mathbf{x}}) \geq 0, & i \in I \end{cases}$$

Mathematical Formulation

The **Optimization Problem**

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} f(\bar{\mathbf{x}}) \quad \textit{subject to} \quad \begin{cases} c_i(\bar{\mathbf{x}}) = 0, & i \in E \\ c_i(\bar{\mathbf{x}}) \geq 0, & i \in I \end{cases}$$

Here E is the set of **equality constraints**, and I the set of **inequality constraints**.

Note that a maximization problem can be converted into a minimization problem:

$$\max_{\bar{\mathbf{x}} \in \mathbb{R}^n} f(\bar{\mathbf{x}}) \quad \Leftrightarrow \quad \min_{\bar{\mathbf{x}} \in \mathbb{R}^n} [-f(\bar{\mathbf{x}})]$$

and a less-than-or-equal-to constraint can similarly be converted into a greater-than-or-equal-to constraint.

Feasible Region

The set of all $\bar{\mathbf{x}} \in \mathbb{R}^n$ which satisfy the constraints $\bar{\mathbf{c}}$ is called the **feasible region**, *e.g.* if

$$\begin{array}{rcrcr} c_1(x_1, x_2) & = & x_1^2 + x_2^2 & \geq & 1 \\ c_2(x_1, x_2) & = & -(x_1^2 + x_2^2) & \geq & -4 \end{array}$$

then the feasible region is the annulus:

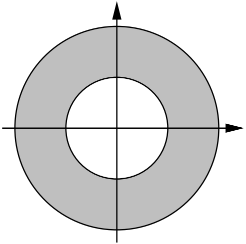


Figure: *The annulus* $1 \leq r \leq 2$.

Constrained vs. Unconstrained Optimization

Optimization problems of the form

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} f(\bar{\mathbf{x}}) \quad \textit{subject to} \quad \begin{cases} c_i(\bar{\mathbf{x}}) = 0, & i \in E \\ c_i(\bar{\mathbf{x}}) \geq 0, & i \in I \end{cases}$$

can **be classified according to the nature of the function and constraints** (linear, non-linear, convex, etc.) — the key distinction is between problems that have constraints, and problems that do not:

Constrained Optimization Problems: arise from models that include explicit constraints on the variables. They can be relatively simple, or nasty non-linear inequalities expressing complex relationships between the variables.

Constrained vs. Unconstrained Optimization

Unconstrained Optimization Problems arise directly in some applications; if the constraints are "natural" it may be safe to disregard them during the solution process and verify that they are satisfied in the solution.

Further, constrained problems can be restated as unconstrained problems — the constraints are replaced by penalizing terms in the objective which "discourage" violation of the constraints.

The more complicated the constraints, the more difficult it is to find the optimal solution. The absence of constraints is the easiest case.

Continuous vs. Discrete Optimization

In many applications, the variables ($\bar{\mathbf{x}}$) can only take integer values — very few people would be interested in buying 3/4 of a TV set, or receive 1/3 of a package; the electrons in an atom can only exist in certain quantum states, etc, etc.

The Discrete Optimization Problem is harder than the Continuous Optimization Problem. — One way to get "close" to solving the discrete problem is to solve the problem as if it is continuous and then round or truncate the solution to integer values. This will often give a sub-optimal integer solution and/or a solution that is **infeasible**.

Here we will only consider the "easier" Continuous Optimization Problem.

Global and Local Optimization

The fastest optimization algorithms seek a **local solution** — a point where the objective is smaller than all other feasible points in its vicinity.

The best solution — the **global minimum** is usually hard to find, but is often desirable.

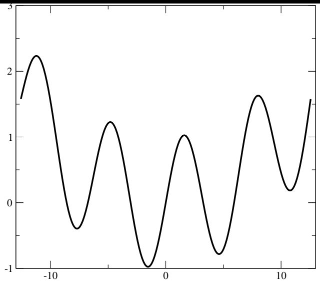
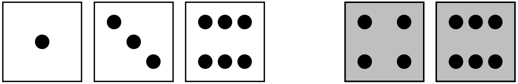


Figure: *A function with multiple local minima, and one global minimum.*

Under certain circumstances (*e.g.* see convexity) there is only one minimum.

We will focus on local optimization algorithms, but note that (most) global algorithms will solve a sequence of local optimization problems.

Stochastic and Deterministic Optimization



In many applications it is impossible to fully specify all parameters at the time of formulation; in quantum physics, the stock market, or the game of "risk" some quantities are "random" and are best modeled using some probability model.

We will focus on **deterministic optimization** problems, where the model can be fully specified when we formulate the problem.

However, in many cases the solutions to **stochastic optimization** problems are formulated as sequences or collections of deterministic problems.

Convexity: Definitions

There are two "types" of convexity which impact optimization problems:

- $S \subseteq \mathbb{R}^n$ is a **convex set** if the straight line segment connecting any two points in S lies entirely inside S. Formally, for any two points $\bar{\mathbf{x}} \in S$ and $\bar{\mathbf{y}} \in S$, we have $(\alpha\bar{\mathbf{x}} + (1-\alpha)\bar{\mathbf{y}}) \in S$ for all $\alpha \in [0,1]$.

- f is a **convex function** if its domain is a convex set and if for any two points $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ in this domain, the graph of f lies below the straight line connecting $(\bar{\mathbf{x}}, f(\bar{\mathbf{x}}))$ to $(\bar{\mathbf{y}}, f(\bar{\mathbf{y}}))$ in $\mathbb{R}^{n+1}$. That is, we have

$$f(\alpha\bar{\mathbf{x}} + (1-\alpha)\bar{\mathbf{y}}) \leq \alpha f(\bar{\mathbf{x}}) + (1-\alpha)f(\bar{\mathbf{y}}), \quad \forall \alpha \in [0,1]$$

Convexity: Illustrations

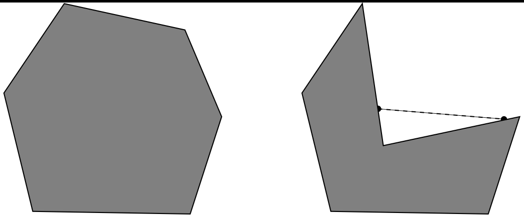


Figure: *A convex (left) and a non-convex set (right) in* $\mathbb{R}^2$.

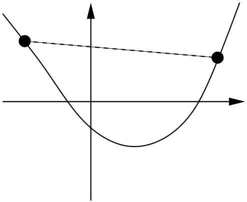


Figure: *A convex function.*

Convexity: Notes

A function f is said to be **concave** if $-f$ is convex.

Optimization algorithms for unconstrained problems are usually guaranteed to converge to a stationary point (maximum, minimum, or inflection point) of the objective f.

If f is convex, **then** the algorithm has converged to a global optimum.

`Bottom line:` Convexity simplifies the problem.

Summary

Easier	**Harder**
Unconstrained	*Constrained*
Continuous	*Discrete*
Local Optimization	*Global Optimization*
Deterministic	*Stochastic*
Convex	*Non-Convex*

Table: *Summary of some factors impacting the difficulty of the optimization problem.*

In this class we will mainly look at Local Optimization methods for Deterministic, Unconstrained, Continuous, Convex functions over Convex sets. Still, it will be a challenging semester!

Algorithms

Optimization algorithms are **iterative** and generate a sequence of successively better estimates of the solution.

Three key attributes characterize each (good) algorithm:

`Robustness:` Algorithm performance on a wide variety of problems (of the same type), for a range of reasonable choices of initial values.

`Efficiency:` We prefer fast algorithms that do not require excessive amounts of storage.

`Accuracy:` The algorithm should find the solution without being overly sensitive to errors in the data or roundoff errors in the computations.

These goals are often **conflicting** — hence careful consideration of trade-off between the goals is a key part of this course.

Math 693A: Introduction — What we will cover

- `Topic-1` Unconstrained Optimization
- `Topic-2` Line Search Methods
- `Topic-3` Trust Region Methods
- `Topic-4` Conjugate Gradient Methods
- `Topic-5` Quasi-Newton Methods
- `Topic-6` Large-Scale Unconstrained Optimization
- `Topic-7` Calculating Derivatives: Automatic differentiation
- `Topic-8 & 9` Least Squares Problems; Nonlinear Equations
- `Topic-10` Introduction to Artificial Neural Network

Index

Course Literature, 10
Grading
- Homework, 12
- Late policy, 23
- Project, 12

Matlab
- Free download, 33

Optimization
- Mathematical formulation, 38

Prerequisites, 34