# ME/ECE/MATH 497/597: Optimization Theory and Practice

Fall 2023

| | | | |
|---|---|---|---|
| **Instructor:** | Aykut C. Satıcı | **Time:** | TT 10:30 – 11:45 a.m. |
| **Email:** | aykutsatici@boisestate.edu | **Place:** | MEC 106 |
| **Room:** | RUCH 233 | **Office hours:** | TT 12:00 – 1:00 p.m. |

**Course Page:**

- Class website: `https://canvas.boisestate.edu`
- Slide sets:
  - Current: `https://symplectomorphism.github.io/opt-publish`
  - All: `https://github.com/symplectomorphism/opt-publish`
- Youtube videos: `https://tinyurl.com/optimization2023`

**Office Hours:** Tuesday and Thursday 12:00 p.m. – 1:00 p.m. @MEC 103, or by appointment.

**Prerequisites:** Students of this class will benefit from a working understanding of

- `Calculus` (MATH 170/175) is a foundational requirement as we will liberally take derivatives and integrals.
- `Linear algebra` (MATH 301/503) will be used extensively as we study, vectors, gradients, hessians.
- `Programming` (CS 111/117/121) (Python, Julia, C/C++) will be used to set up and solve large programming assignments and/or projects.

**Course Description:** This course will equip students with the basics of optimization theory, numerical algorithms and applications. The course is divided into three main parts: linear programming (simplex method, duality theory), unconstrained methods (optimality conditions, descent algorithms and convergence theorems), and constrained minimization (Lagrange multipliers, Karush-Kuhn-Tucker conditions, active set, penalty and interior-point methods). Applications in engineering, operations, finance, statistics, etc. will be emphasized. Students will use Julia or Python's optimization packages to obtain practical experience with the material.

**Learning Outcomes:**

- Recognize, formulate, and solve linear programming problems.
- Understand the simplex method for linear programming.
- Be able to apply linear programming to applications such as network flow, game theory, finance, and machine learning.
- Learn steepest descent, Newton's method, and conjugate gradients for unconstrained optimization.
- Understand the necessary and sufficient conditions for unconstrained and constrained optimization, including Lagrange multipliers and the KKT conditions.
- Understand active-set and penalty/barrier methods for constrained optimization.
- Be able to apply numerical packages to solve optimization problems.

**Texts and Materials:** We have the following main textbook.

- E. Chong and S. Zak, *An Introduction to Optimization*, $4^{\text{th}}$ edition, Wiley, 2013.

There are several other excellent robotics books in the literature, some of which I have listed below in no particular order. At times, I will take some material from some of these resources.

- D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, $4^{\text{th}}$ edition, Springer, 2015.
- D. Bertsekas, *Nonlinear Programming*, $3^{\text{rd}}$ edition, Athena Scientific, 2016.

**Grading Policy:** Team project 1 (25%), Midterm exam (20%), Team project 2 (25%), Final project (30%).

- Several homework assignments will be assigned throughout the semester. Students are encouraged to tackle these questions for a 5% bonus.
- The homework questions will not be rigorously graded; I will merely look them over to see if the student has given the questions a decent amount of thought and if so will award the full grade.
- The team projects will be completed in *two weeks* by teams that consist of at most two students. Each team will submit a report by the end of the two weeks.
- The final project can be performed by teams of at most two students. This project must be identified by the team members and should ideally help the students with their research projects. A report is due by the last date of course instruction. In addition, each team will present their results during the last week of course instruction (approximately 10-15 minutes).
- You are responsible for all the information given in class verbally and/or in writing. Any information about the course on the web may be superseded by the information given in the class.

**Submission Guidelines:**

- Submit over Canvas as a single PDF or as a single compressed file if there are multiple files.
- Whenever you are submitting a program, please provide instructions on how to execute it in a `README` file. Note that, this means you have to make sure your program will work on **my** computer. Program defensively!

**Important Dates:**

Team Project 1 ........ (start date) September 19, 2023
Midterm Exam ....................... October 12, 2023
Team Project 2 .......... (start date) October 26, 2023
Final Project ..... (report due date) December 07, 2023

**Course Policy:**

- For written work, please write legibly and present your work neatly.
- Your solution to the problem must follow a logical progression.
- Use the submission guidelines above for anything that you submit in this class.

**Class Policy:**

- Class participation and cooperation among students are highly encouraged.
- Student feedback will be collected throughout the semester and adaptation will be undertaken.
- Extra lectures or problem solving sessions may be scheduled when necessary.

**Project Guidelines:**

- A couple of projects will be assigned throughout the semester that will involve solving a real-world optimization problem to be solved by teams consisting of at most two students.
- The solution to these problems will have theoretical, computational aspects.
- The computational part will need to be implemented as a computer program as it will be impossible to solve by hand.
- I can hold my own in many programming languages so you can submit your programs in many different languages as well.
  - I recommend using one of: Python or Julia.
  - If you are feeling confident and want an extra challenge you are free to use: Rust or C++.
  - I am ok with submissions in Matlab.
  - If you want to code in JavaScript or TypeScript, I am fine with that, but please provide an extra thorough `README` file as I will start learning this programming language if/when you submit.
  - Unmentioned programming languages must *not* be used.
- You **must** provide a `README` file that explains exactly how I can run your program and inspect your results.

**Academic Honesty:** Academic dishonesty is **NOT** tolerated. Students are responsible for reading and understanding the student code of conduct. For more information, click here.

**Make-up Policy:** Students are eligible for a make-up exam if they miss a scheduled exam due to a death in the immediate family, serious illness or military service.

- Students are responsible for informing the instructor of their excused absence as soon as possible and their excused absence must be *officially verified*.
- If your excuse is acceptable, you will be given a single *comprehensive* make-up exam, which will be delivered during the last week of the classes.

**Disabilities Policy:** Students with disabilities needing accommodations to fully participate in this class should contact the Educational Access Center.

**Tentative Course Outline:**

Linear Programming

Week 1 – 2: Basic Properties of Linear Programs

- Examples of LP problems
- Basic feasible solutions
- The fundamental theorem of LP
- Relations to convex geometry
- Farkas's lemma and alternative systems

Week 3 – 4: Duality and Complementarity

- Dual LP and interpretations
- The duality theorem
- Geometric and economic interpretations
- Sensitivity and complementary slackness
- Selected applications of duality

Week 5 – 6: The Simplex Method

- Extreme points
- The primal simplex method
- The dual simplex method
- Efficiency analysis of the simplex method

Week 7: Interior-Point Methods

- The analytic center
- The central path
- Solution strategies
- Termination and initialization

Unconstrained Problems

Week 8 – 9: Basic Properties of Solutions and Algorithms

- First-order necessary conditions
- Examples of unconstrained problems
- Second-order conditions
- Convex and concave functions
- Minimization and maximization of convex functions
- Global convergence of descent algorithms

Week 10 – 11: Basic Descent Methods

- Line search algorithms
- The method of steepest descent: first-order
- Multiplicative steepest descent
- Newton's method: second-order
- Coordinate and stochastic gradient descent methods

Constrained Optimization

Week 12 – 13: Constrained Optimization Conditions

- Constraints and the tangent plane
- First-order necessary conditions (equality constraints)
- Second-order conditions (equality constraints)
- Inequality constraints
- Lagrangian duality and zero-order conditions
- Rules for constructing the Lagrangian dual explicitly

Week 14: Primal Methods

- Infeasible direction and the steepest descent projection method
- Feasible direction methods: sequential linear programming
- The gradient projection method