Syllabus for Math 485, Modern Optimization Algorithms

Instructor: Dr. Zev Woodstock
Office: Roop 310
Office hours: Posted on Canvas

woodstzc@jmu.edu
zevwoodstock.github.io
Canvas link

Prerequisites: Programming (e.g., math 248), linear algebra (math 300/238/434), and calculus (236/237).

Class Details: Math 485(Section 0001), Modern Optimization Algorithms
M/W/F: 11:30-12:20 in Roop 105

Optional books No books are required for this course. The class notes will be self-contained, and they will draw from a variety of recent publications within the optimization literature in addition to the listed book references below. For students interested in further reading on the topic, please use the following reference books that cover some of the material from this course.

→ *First-Order Methods in Optimization*, A. Beck, Society for Industrial and Applied Mathematics, 2017.

→ *Large-scale convex optimization: algorithms & analyses via monotone operators*, E. Ryu and W. Yin, Cambridge University Press, 2022.
preprint: `https://large-scale-book.mathopt.com/LSCOMO.pdf`

→ *Conditional gradient Methods: From core principles to AI applications*, G. Braun, A. Carderera, C. W. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, and S. Pokutta, MOS-SIAM Series on Optimization, 2025.
preprint: `https://arxiv.org/pdf/2211.14103`

→ *Convex analysis and monotone operator theory in Hilbert spaces* (2nd ed.), H. H. Bauschke and P. L. Combettes, Springer, 2017.

→ *Variational Analysis*, R. T. Rockafellar and R. J-B Wets (3rd printing), Springer, 2009.
preprint:
`https://sites.math.washington.edu/ rtr/papers/rtr169-VarAnalysis-RockWets.pdf`

Programming language: If you are required to submit code for an assignment, it can be submitted in Python 3, Julia, Octave, or MATLAB (although the entire assignment must be in one language). Python, Julia, and Octave are free. MATLAB usually requires payment, although JMU students can install MATLAB for free. You are not required to install all four languages; please select a language (or languages) that will best prepare you to meet your career goals. If you have not programmed before, you are advised to only select one language. There are *many, many* free online resources for installing these programming languages, and you can get started by following the instructions in the following links Python 3, Julia, Octave, or MATLAB.

PROGRAMMING DOCUMENTATION For programming assignments, this class will accept programs written in `Julia`, `python3`, `octave`, and (if you prefer paid software), `MATLAB`. The documentation for the four languages accepted in this class are below.

Python 3 documentation: `https://www.python.org/doc/`

Julia documentation: `https://docs.julialang.org/en/v1/` PDF version is here.

Octave documentation: `https://docs.octave.org/latest/`

Matlab documentation: `https://www.mathworks.com/help/matlab/index.html`

CALCULATORS: Unless explicitly allowed within an assignment, the only machines allowed for assistance on assessments have **only** the ability to compute $+$, $-$, $\times$, $\div$, exponentiation, logarithms, trigonometric functions, and roots (e.g., square root). Machines with any other capabilities, (e.g., phones) are not allowed. For *homework and programming assignments*, graphing calculators such as Desmos and a programming languages for scientific computing (e.g., Python, Octave, Julia, or Matlab) are allowed. Unless specifically stated as a requirement in an assignment, the use of LLMs (e.g., ChatGPT, LLAMA3, Copilot) or algorithmic solvers (e.g., Wolfram Alpha, Maple, and Gurobi) are prohibited.

GRADING: Final grades will be comprised of 65% homework and 35% for a semester-long project. Homework may include programming for some problems. The semester-long project will be introduced in-class with detailed supporting documentation. Extra credit opportunities may be announced in-class periodically throughout the semester. Grades will be assigned according to the following scheme, with $+/-$ assigned if within 2 points of the closest grade.

A: Over 90%, B: $89.\overline{9}\%$ – 80%, C: $79.\overline{9}\%$ – 70%, D: $69.\overline{9}\%$ – 60%, F: under 60%.

UNIVERSITY POLICIES: For JMU policies for attendance, inclement weather, disability accommodations and religious accommodations, please see: `http://www.jmu.edu/syllabus/`

HONOR CODE: All students must abide by the JMU honor code at all times. Cheating comes with serious repercussions: `http://www.jmu.edu/honorcode/`.

ATTENDANCE: Attendance of all students is expected. Students are responsible for all of the information covered in-class. Absences may be excused with appropriate evidence according to University Policy (see above) in combination with appropriate notice to the instructor. *Excused absences include, e.g.,:* weddings, funerals, family/home emergencies, religious holidays, mental health reasons, physical health reasons, doctor appointments, hospitalizations, jury service and subpoenas, NCAA athletics involvement, attending a visitation event for newly-admitted students to a graduate program, and anything else the Provost's office deems worthy (in writing) of an excused absence.

If an excusable absence is *known in advance* (e.g., court appearance, medical appointment, etc.), notice must be provided via email to the instructor as early as possible prior to the absence (minimum 24 hours). If the student misses class for an *excusable and unexpected* reason (e.g., due to unexpected hospitalization), they must provide notice within 1 week of their return to university activities. The instructor may require evidence of an absence excuse, but per University Policy, students are never obliged to divulge specific medical information.

FIRST WEEK ATTENDANCE POLICY: At the instructor's discretion, any student registered for a class in the Department of Mathematics and Statistics who does not attend at least one of the first two scheduled meetings of the class (or does not attend the first scheduled meeting of a class that meets once a week) MAY be administratively dropped from the class. Students will be notified by email if they will be dropped. Students who fail to attend should not assume they will be administratively dropped by their instructor; it is the student's responsibility to drop the course on their own or they will receive a grade at the end of the semester. All students are responsible for verifying the accuracy of their schedules and changes made in their schedules.

PROGRAMMING ASSIGNMENTS/CLASSWORK/HOMEWORK/QUIZZES: In-class group and individual assignments may be given with little or no notice; if these assignments are missed for an unexcused reason, 0 credit will be given. If a student misses a quiz or classwork with an excused absence (with proper notice in accordance with the Attendance policy above), then their classwork/homework/quiz average will omit the missed assignment (hence it will not negatively impact their grade). If a student has an excusable reason for absence during a period when an out-of-class individual assignment is due, they are eligible for a deadline extension; students must request this extension within one week of their return to university activities. If all assignments in a particular category (e.g., homework) are missed, 0 credit will be received in that category.

For this class, *programming assignments* may be graded based on any (or all) of the following criterion: (1) submission of working code, (2) a technical write-up with documentation explaining how to use a program and how it works, (3) a "stand-up" style short presentation explaining how a program works with a live demonstration, and/or (4) an interview asking students to explain specific lines of their submitted code. If I attempt to run your code and it does not compile/run on my machine, then the maximum possible score for the coding part of your assignment is 50% (even if everything is correct until your final line of code).

EMAILS: All email correspondence must be formatted in a professional manner: this includes a greeting, closing, proper grammar/formatting, and all relevant information (e.g., class, section number, and/or absence dates). Emails that do not abide by this policy will not receive a response. I will not help diagnose malfunctioning code by e-mail: if you are having an issue with your code not working, search the documentation and the internet for a diagnosis / solution.

GRADE APPEALS: Specific grading disputes (points totaled incorrectly, partial credit appeals, missing assignments on Canvas, etc.) should be brought to the instructor's attention within 1 week of return of the assignment or its posting date on Canvas. This does not guarantee a grade change. Appeals outside of this timeframe will not be considered. Appeals must be received before final grades are submitted to the registrar.

ADDITIONAL HELP: Please make use of the following resources available to JMU students with no extra cost.

→ This class relies on some programming. To get extra practice, check out Project Euler.

→ My office hours are on Canvas **and by appointment**. If my office hours do not meet your schedule, write me an email and we will find a meeting time.

→ If class is missed for any reason, refer to the schedule in this syllabus which **section** of the text to read; also, check the class website for additional notes and resources.

→ JMU offers accessibility accommodations, including extra exam time, note takers, and recorded lectures.

→ The JMU Counseling Center has many resources and programs that are **free** to students.

→ Madison Cares is a service for referring students of concern who may need help.

TENTATIVE SCHEDULE

Week 1: 8/26 – 8/28: *Preliminaries :*
Extended real number line, supremum/infemum, optimization terminology, existence of solutions, and inner product spaces
Week 2: 8/31 – 9/4: *Preliminaries; start applications:*
Inner product spaces, adjoints, and Hilbert spaces; applications
Week 3: 9/7 – 9/11: *Applications / modeling optimization problems:*
Classifier training, sparsity, image processing (deblurring, superresolution, compression, . . .)
Week 4: 9/14 – 9/18: *Applications / modeling optimization problems:*
Image recovery (wavelet sparsity, convolution, TV, denoising) recommender system training;
  9/15: Withdrawal deadline
Week 5: 9/21– 9/25: *Theoretical foundations:*
Convexity of functions and sets, projection operators, linear minimization oracles and their complexity
Week 6: 9/28 – 10/2: *Theoretical foundations:*
Differentiability, smoothness, subdifferential calculus
Week 7: 10/5 – 10/6: *Theoretical foundations:*
Subdifferential calculus, stationarity / nonconvexity, proximity operators
  Fall Break: 10/7 – 10/11: no class
Week 8: 10/12 – 10/16: *Algorithmic foundations:*
Mathematical and algorithmic properties of proximity operators; firmly nonexpansive operators

Week 9: 10/19 – 10/23: *Algorithmic foundations:*
Convergence of firmly-nonexpansive algorithms (gradient descent, proximal point algorithm, etc.)
Week 10: 10/26 – 10/30: *Algorithmic extensions:*
Subgradient projectors, families of gradient-descent algorithms (SGD, adagrad, adam, . . .), historical issues with nonsmooth problems, and recent results on O-minimality. Discussion on algorithmic "bells and whistles" including adaptiveness, acceleration, block-iterative, and stochastic variants.
Week 11: 11/2 – 11/6: *Proximal splitting algorithms:*
Forward-backward, Douglas-Rachford, Pierra's product space trick and "minimal-lifting"
Week 12: 11/9 – 11/13: *Finish proximal splitting algorithms; start Frank-Wolfe algorithms*
Application of DR in product space for SVM training; geometry of the Frank-Wolfe algorithm
Week 13: 11/16 – 11/20: *Frank-Wolfe / conditional gradient algorithms:*
algorithmic structure, stepsize choices, splitting variants, convergence proofs
Thanksgiving break: 11/21 – 11/29: no class
Week 14: 11/30– 12/4: *Frank-Wolfe / conditional gradient algorithms:*
Finish convergence proofs; start student presentations*
Week 15: 12/7 – 12/11 Student presentations*
Final presentations:*[1]: Wednesday, 12/16/2026, 10:30 AM - 12:30 PM in Roop 105

*: Student presentations will cover topics including, but not limited to, Newton-type methods, BFGS/L-BFGS, Lagrange / KKT theory, fixed-point theory, integer-constrained optimization, branch-and-bound, simulated annealing, 0th-order methods, and benchmarking academically-licensed solvers.

[1] https://www.jmu.edu/registrar/wm_library/spring_exam_schedule.pdf