MATH 317: Operations Research: MWF 10-10:50am

Professor Steven Miller (sjm1@williams.edu), Bascom 106D (617-835-3982)

My office hours: TBD and whenever I'm in my office (click here for my schedule)

COURSE DESCRIPTION: In the first N math classes of your career, you are sadly misled as to what the world is truly like. How? You're given exact problems and told to find exact solutions. The real world is sadly far more complicated. Frequently we cannot exactly solve problems; moreover, the problems we try to solve are themselves merely approximations to the world! We are forced to develop techniques to approximate not just solutions, but even the statement of the problem! In this course we discuss some powerful methods from advanced linear algebra and their applications to the real world, specifically linear programming and if time permits random matrix theory.

Format: Evaluation based on homework, presentations and write-ups, class participation and exams.

Prerequisites: Linear algebra (programming experience and analysis are desirable, but not necessary)

CONTACTING ME: You can reach me in Bascom 106D (if I'm there it's office hours), email sjm1@williams.edu, or ***anonymously through ephsmath@gmail.com (password 1793williams; sadly with new securities may not work).***

OBJECTIVES: The goal is to introduce you to advanced concepts and problems in linear algebra, specifically linear programming, with an emphasis on mathematical modeling. There will be numerous opportunities to work on real world problems for companies. This course in previous iterations has real analysis as a pre-req, but not this version. We will move at a fast pace at times, and you are responsible for doing a significant amount of reading on your own.

TEXTBOOK/SYLLABUS: The textbook is Mathematics of Optimization: How to do things faster; I also urge you to get Methods of Mathematical Economics: Linear and Nonlinear Programming, Fixed-Point Theorems by Joel N Franklin.

GRADING POLICY: Homework: 15%, Class Presentation: 5%, Midterm 40%, Final or Project: 40%. (Note the grading percentages may change a bit. A large portion of your work and grade in this class will come from a group project or a final, to be determined early in the semester, if a project then it will include a write-up and a class presentation.)

COURSE DISCLAIMER: I may occasionally say things such as `Probability is one of the most useful courses you can take' or 'If you know probability, stats and a programming language then you'll always be able to find employment'. I should say `you should always be able to find employment', as nothing is certain. Thus, please consider yourself warned and while you may savor the thought of suing me and/or Williams College, be advised against this! I'm saying this because of the recent lawsuit of a graduate didn't have a job, and sued her school!

Below is a more detailed description of the course. The content is still a bit in flux as I adjust based on the interest and background of the class. The topics below are taken to give students a broad sense of issues in efficiency and optimization, both in the need to do it and how it is done.

1. **Ancient / Classical algorithms:** These are warm-ups meant to show students how to do common items faster (i.e., things they've seen many times). Examples include the following.

a. **Babylonian multiplication:** base 60 is a pain; instead of learning the 60x60 table for xy, they noted $xy = ((x+y)^2 - x^2 - y^2) / 2$, reducing to just needing to know squares, subtraction and division by 2. This leads to the concept of a look-up table and interpolation, and is a great opening to talk about classical tables (for special functions), and how modern appliances (such as cell phones) work with limiting processing power.

b. **Horner's algorithm:** fast polynomial evaluation by cleverly grouping parentheses. We will discuss why such efficiencies are needed. One project will be for students to write code to iterate polynomials and create fractal sets, with and without Horner's algorithm, noticing the sizable decrease in run-time (or at least I hope they will see this – I saw that in the 90s when I wrote such code!).

c. **Fast exponentiation:** The method of repeated squaring to evaluate multiplication quickly. This will lead into discussions of the advantages of different bases and applications in cryptography, and issues in increased storage in some implementations (and ways to minimize that). Student will write code to perform these computations.

d. **Strassen algorithm:** Will discuss how to obtain a power improvement and what this means, fits nicely with the Horner algorithm discussion. Will discuss real world implications of being able to do matrix multiplication faster. Will talk about how things are easier to program when the matrix size is a power of 2, and what that means in determining how data is collected. If time permits we will discuss algorithms involving sparse matrices.

2. **Efficiency Revisited:** These are some important problems where students may not have had as much experience. The point is to look at a variety of problems where standard definitions or brute force are too slow.
a. **Finding eigenvalues:** It should be a prosecutable crime to believe that one finds eigenvalues by looking at determinants! We will discuss a variety of methods (some involving the method of repeated squaring) to compute these efficiently. Students will write code to numerically approximate eigenvalues.

b. **Solving difference equations:** While in principle the definition and initial conditions determine all the terms, there is frequently a need to jump far down in the sequence or investigate limiting behavior. This leads to discussions on the Method of Divine Inspiration (to guess solutions to these, and related problems in differential equations), and generating functions. We will discuss recent work of myself and my students on Zeckendorf decompositions (writing any number as a sum of non-adjacent Fibonacci or, more generally, other recurrence relations with some constraints), and numerically investigate properties of the number of summands (especially the distribution of their gaps).

c. **Binomial Coefficients:** Pascal's triangle modulo 2 is a beautiful introduction to fractals. We will study this system and discuss the difficulties in efficiently computing it. We will analyze several approaches, and students will be required to write code to create. Issues will involve data storage, data access, and jumping to compute the triangle at a certain spot. We will also discuss issues involved in graphically presenting the results, and efficient ways to computing binomial coefficients and prove identities. For example, if we need all the coefficients in a given row, it is absurd to compute them individually, even though we have a closed form expression for each. It is far better to notice that (n choose k+1) = (n choose k) x ((n-k)/(k+1)). If time permits we will discuss computations related to graphing Ulam's spiral.
d. **Sorting algorithms:** If time and interest permit, we'll discuss various sorting algorithms as these are great examples to introduce run-time complexity calculations, and issues with how we score an algorithms (are we concerned about average run-time, worse case, …), and the advantages of switching off between two algorithms.

e. **Factorization / primality testing:** Interestingly it's possible to efficiently determine if a number is composite *without* finding any factors! We'll discuss some of the issues with factoring and primality testing, learning a small amount of group theory as needed. Students with stronger backgrounds will be encouraged to read the "Primes in P" paper. We will see applications of fast exponentiation with Fermat's little theorem.

3. **Linear Programming:** The highlight of the course is a detailed analysis of linear programming. We will discuss the history of the subject, from original papers that claimed to be of theoretical interest but no practical value as there would never be any way to do so many computations, to its recent successes and challenges. Students will be required to write or use code to solve a linear programming problem.
a. **Basics:** basics of linear programming, duality, standard examples (diet problem).

b. **Simplex Method:** We will prove the simplex method works, and analyze its run-time as much as possible. This will be the theoretical highpoint of the semester.

c. **Binary Integer Programming:** While we currently do not have an algorithm capable of resolving binary integer programming problems quickly, we can often get close to optimal very quickly, which suffices for many purposes. This is especially true as we frequently do not know what values the parameters take! This is the part of the course that will have the greatest stats component. We will discuss a paper I wrote with marketing colleagues which solved a binary integer programming problem for the movie industry, and discuss the issues with parameter estimation and model formation. This, as well as factorization, are excellent opportunities to discuss P – NP.

d. **Applications (Scheduling, Stochastic):** We will study progress on the Traveling Salesman Problem and scheduling problems. We will explore how variations in parameter values effect the solutions. We will study stochastic programming problems, where the parameters themselves are independent random variables (which will require a quick crash course in probability, and hopefully motivate students to take that class, if they haven't done so, in the spring). Another interesting application is using linear programming to compute what elimination numbers ***should*** be in baseball (MLB incorrectly defines the term and often, as in 2004, incorrectly calculates them!). Other applications include Hales' proof of Kepler's conjecture on sphere packing, Support Vector Learning, Quadratic Programming, ….

The remainder of the course will be advanced special topics chosen in part from the interests of the class. Possible topics include fair division, Nash equilibria and fixed point theorems, and matching algorithms (such as Gale-Shapley).

Here are the links to the first few days of the class the last time I taught it (2016)

- **Week 1: Sept 9, 2016**
  - Welcome slides for first class.
  - Handout for first class.
  - Review your linear algebra: review lectures online at thttp://web.williams.edu/Mathematics/sjmiller/public_html/linprog/videos/LinProg_Lect01_LinAlgReview.MP4 and http://web.williams.edu/Mathematics/sjmiller/public_html/linprog/videos/LinProg_Lect02_LinAlgReview.MP4
  - Start exploring mathematical software (R, Mathematica). Learn LaTeX. I have handouts and videos: http://web.williams.edu/Mathematics/sjmiller/public_html/math/handouts/latex.htm
  - Homework: To be emailed to me by noon on Sunday, September 11: Email me a short note (a paragraph suffices) on what you want to get out of this course and what lesson you learned from the graduation speech by Uslan (http://www.graduationwisdom.com/speeches/0018-uslan.htm). Also send me a resume/CV (send as a pdf, with a good filename -- CV.pdf or resume.pdf is a sure way to make me angry).
  - Homework: To be done by start of class on Monday, September 12: Read Chapter 1.
  - Homework: To be done by start of class on Friday, September 16: find something in your life that you can do better / more efficiently, and tell me about it (in person or email). You should constantly be thinking about how to optimize (but please think about what you choose to share!).
- **Week 2: Sept 12 to 16, 2016:**
  - Review your linear algebra (see https://www.youtube.com/watch?v=f7ii-sjfBrA&feature=youtu.be and http://youtu.be/yGVq-zx6Whg).
  - Read ahead in the course book and my notes -- I'll be lecturing on a hodgepodge of topics you have hopefully seen before, but casting them in a new light. Use this time to make sure your linear algebra is strong and build a strategic reserve of material read for later.
  - Homework problems: Due Monday, September 19: #0: Write a program to generate Pascal's triangle modulo 2. How far can you go? Can you use the symmetries to compute it quickly? You do not need to hand this in. From the textbook: #1: Exercise 1.7.4 (there are many trig tables online: see for example http://www.sosmath.com/tables/trigtable/trigtable.html), and read BUT DO NOT DO Exercise 1.7.5. #2: Exercise 1.7.7. #3: Exercise 1.7.18. #4: Exercise 1.7.34. #5: Exercise 1.7.26. #5: Exercise 1.7.36.

Objectives for Math 317: Introduction to Operations Research

The following is entirely optional, but describes some of my thoughts about the course, ranging from why it is structured the way it is to what I want you to get out of this course in particular and your education in general. When I was in high school, I remember the Boston Globe ran an article where they asked numerous 'famous' people in the state: what 10 books should **every** high school student read? The answers were, for the most part, disappointing. The English professor had ten literary selections, ranging from some Shakespeare and Milton to stuff I can no longer remember. Most of the others had lists greatly biased towards the small part of the world they studied; very few had balanced lists that would help prepare the general student for the world (Governor Weld was one of the few who did).

I **strongly** urge you to view the following clip on YouTube: Did You Know (2009 version). Some of the more interesting statistics / quotes:

- Preparing students for jobs that don't exist using technologies that haven't been invented to solve problems we don't know are problems yet.
- We are living in exponential times:
  - 31 billion searches on google each month now; it was 2.7 billion in 2006.
  - The first text message was in 1992; more are sent today than the population of the planet.
  - Number of internet devices: 1,000 in 1984, 1,000,000 in 1992, 1,000,000,000 in 2008;
  - Estimated that 4 exabytes (4×10^{19} bytes) of unique information will be generated this year, which is more than has been generated in the past 5000 years!
  - Amount of new technical information doubles every two years.

Another great clip to listen to is the TED lecture of Malcolm Gladwell on spaghetti sauce. The main point of this lecture is

- Important to ask the right question; what you think is the right question frequently isn't. I won't do the lecture justice by summarizing, so here's a tantalizing tidbit: this surprising question led to Prego making $600 million in 10 years on extra chunky spaghetti sauce.
- There are lots of great clips on TED; another one of my favorites is Dan Pink on Motivation. Some very interesting observations here on how to create an optimal environment for creativity to flourish.

What does all this have to do with our class? The point is that it is hard to predict what will be useful to most of you in your career(**s**) after Williams (save for the few who go off to grad school to study number theory, where I can do a pretty good job predicting). It is imperative that you learn to be problem solvers. The content of a course matters; you need to learn the language, the basic facts, the key theorems, et cetera. But, at least as important, you need to learn how to use these on 'new' problems.

One of my biggest epiphanies as an educator was when I prepared my lecture notes for Math 209 (differential equations). This was the first class I'd ever taught as a college professor which I'd actually taken as an undergrad. I was looking through my old course notes, trying to decide what to include, when I noticed that we did the Bessel equation and function when I was a student. I was shocked; I had no memory of having done this, but I use the Bessel function in about a quarter of my number theory papers. What's the takeaway? You're going to forget much of the material you learn. That's to be expected. Hopefully you'll be able to re-learn it as needed / you'll know where to go to read more about it / you'll have some familiarity that such results exist. I had to relearn the Bessel function in grad school (and did, it's not that bad). What is more permanent is the techniques and methods. These are the things you'll use again and again. They can range from learning how to multiply by 1 or add 0 (two of the most difficult things to do well in mathematics!) to how to count something two ways to how to model the key features of a problem.

One of the goals of this course is to help you become problem solvers. The problems will come analysis and linear algebra, but the methods and techniques, the mindset, should hopefully be applicable to a variety of problems. If you are taking an intro calculus class and Section 3.2 is on the Chain Rule, it's a safe bet that you should use the chain rule to solve problems from that section. The real world (or advanced academia) is not like that -- you frequently do not know what the 'right' way is to attack a problem (especially open problems that have stumped people for years). This is one of the reasons why I want you to create (and if possible solve!) your own homework problems. The act of creation is a huge part of research. **Most** math papers are **mostly** trivial (or, as I often say, trained monkey work). What does this mean? It means that over 95% of most papers is just straightforward manipulation of previous results (the further you go in math, the more things become straightforward). The most important parts of papers are usually the following two items: (1) asking the right question; (2) coming up with a novel way to attack a problem. Often once the question is asked and the method chosen, the paper writes itself; however, it is very hard at times to ask the right question, or to see a good way to attack it. (As an example, when I taught at Princeton years ago I wrote a handout for my students on how to prove the Fibonacci numbers satisfy Benford's law of digit bias. I decided to try and publish it, and did some research. I found a paper from the 1970s that was almost identical to mine -- basically same theorems and lemmas in the same order! This isn't too surprising, as once the question was asked, this truly was the most 'natural' way to go.)

There are remarkable connections between what seem at first very disparate branches of learning; if you are one of the first people to see such a connection, you have the potential of making a real breakthrough. I **strongly urge** you to tell me what you're interested in. I'll see if I can work it into the course (either in the lectures proper or in the additional comments). The more you explore, the more likely it is you can make one of these great connections. I've been fortunate enough to make some connections between nuclear physics and baseball, and between number theory and tax fraud. Because of this I've had a private tour of Petco Park (where the San Diego Padres play) and given a talk at the Boston headquarters of the IRS (with district attorneys, auditors, and secret service agents in the audience) and been interviewed by the Wall Street Journal about fraud in the recent Iranian elections.

Finally, here is a great website with 10 excellent commencement speeches. It's worth the time reading these; I particularly liked the one by Uslan (on how it's not enough to just have a good idea, but how to get noticed).