CBE 20258 - Numerical and Statistical Analysis

Syllabus

Click here to see the course syllabus.

Office Hours

Instructor

- **D. T. Leighton**
  Office hours are 5-6 pm T, 4-5 pm W, or by appointment. Room 179 Fitzpatrick Hall.
  email: dtl@nd.edu

Teaching Assistants

- Ken Newcomb
  Office hours: TBA
  email: Kenneth.Newcomb.8@nd.edu
- Yang Song
  Office hours TBA
  email: Yang.Song.44@nd.edu
- Prateek Mehta
  Office hours TBA
  email: Prateek.Mehta.10@nd.edu

On-Line Resources

- The notes from last year's version are located here.
- The notes and website from a previous iteration of this course are given here.
- The Matlab Online Documentation website is located at http://www.mathworks.com/help/matlab/index.html. This website provides information on all matlab functions, with useful examples.
- To get help on any command, just type "help" at the command prompt in matlab, followed by the command you are interested in. To see the detailed documentation of any function, just type "doc" followed by the function name.

Class Notes

- Jan. 12, 2016 - What's Numerical & Statistical Analysis all about?
- Jan. 14, 2016 - Numerical Errors
- Jan. 19, 2016 - Systems of Equations
- Jan. 21, 2016 - PLU Factorization
- Jan. 26, 2016 - QR Factorization

- Jan. 28, 2016 - Singular Value Decomposition
- Feb. 02, 2016 - Elementary Statistics and Probability
- Feb. 04, 2016 - Error Propagation
- Feb. 09, 2016 - Hypothesis Testing
- Feb. 11, 2016 - Linear Regression and the Normal Equations
- Feb. 16, 2016 - Regression Error and Analysis
- Feb. 18, 2016 - Weighted Regression and Analysis of Residuals
- Feb. 23, 2016 - Non-Linear Regression
- Feb. 25, 2016 - Undersampling and the Bootstrap
- Mar. 01, 2016 - Root Finding in Non-Linear Equations
- Mar. 03, 2016 - Mid-Term Exam
- Mar. 08, 2016 - Mid-Semester Break
- Mar. 10, 2016 - Mid-Semester Break
- Mar. 15, 2016 - Systems of Non-Linear Equations
- Mar. 17, 2016 - One-Dimensional Optimization
- Mar. 22, 2016 - Multi-Dimensional Optimization
- Mar. 24, 2016 - Constrained Optimization
- Mar. 29, 2016 - Numerical Quadrature
- Mar. 31, 2016 - Gaussian Quadrature
- Apr. 05, 2016 - Adaptive Quadrature
- Apr. 07, 2016 - Multidimensional Quadrature and Mapping
- Apr. 12, 2016 - Ordinary Differential Equations
- Apr. 14, 2016 - Numerical Stability and Implicit Integration Methods
- Apr. 19, 2016 - Adaptive Step Size Algorithms
- Apr. 21, 2016 - Matrix Methods and Boundary Value Problems
- Apr. 26, 2016 - Partial Differential Equations and Sturm-Liouville Boundary Value Problems

---

Tutorials

This page contains the scripts, functions, and data sets which will be used in the CBE 20258 tutorials.

Homework Assignments

These are links to the homeworks organized by due date. We will be alternating assignments developing your algorithm skills (e.g., how you approach problems) and your programming skills (e.g., implementation of the algorithms). Click on a highlighted date to access the homework due that day.

- Algorithms 1, Jan 21, 2016 (solution).
- Project 1, Jan 28, 2016 (solution).
- Algorithms 2, Feb 04, 2016 (solution).
- Project 2, Feb 11, 2016 (solution).
- Algorithms 3, Feb 18, 2016 (solution).
- Project 3, Feb 25, 2016 (solution).
- Algorithms 4, Mar 17, 2016 (solution).
- Project 4, Mar 24, 2016 (solution).
- Algorithms 5, Mar 31, 2016 (solution).
- Project 5, Apr 07, 2016 (solution).

- Algorithms 6, Apr 14, 2016 (solution).
- Algorithms 7, Apr 21, 2016 (solution).

- Final Project, No later than 6pm, Apr 27, 2016 (solution).

Examinations

- Mid Term Exam Thursday Mar 05, 2016 9:30-10:45 AM
- Final Exam Tuesday May 3, 2016 10:30-12:30
- Past Exams

Back to **ND Home Page** Back to **CBE Home Page**

David.T.Leighton.1@nd.edu

CBE20258 - Numerical & Statistical Analysis

Schedule

- **Class**

  TR 9:30-10:45AM. Room 155 DeBartolo

- **Tutorials**

  Five sections, 303 Cushing

Instructor

- D. T. Leighton

  Office hours are 5-6 pm T, 4-5 pm W, or by appointment.
  Room 179 Fitzpatrick Hall

Textbooks

In addition to the extensive online course notes, and on-line help, the required text for the course is King and Mody, Numerical and Statistical Methods for Bioengineering (2011).

Course Outline

1. Introduction to numerical computation.
2. Representing numbers.
3. Linear equations.
4. Statistics and data analysis.
5. Nonlinear equations.
6. Optimization
7. Quadrature.
8. First-order ordinary differential equations.
9. Higher-order ordinary differential equations.
10. Eigenvalue problems.

Grading

The course consists of four components: Weekly algorithm assignments and projects (cumulative), a concluding final project, a mid-term exam and a final. Attendance at the weekly tutorial (essentially a lab session) is mandatory without a written excuse. Students are expected to have completed the weekly assigned reading and to have outlined their approach to the weekly assignment -prior- to the tutorial. The tutorial may be used to work on the assignment, although getting started early is fine too! The exams will be closed book, in class exams based primarily on the algorithms discussed in the lectures, and will focus on error analysis and statistics. The final project will be substantial in nature, and will synthesize several of the numerical solution techniques developed in class. The cumulative homework and mid-term will count 25% each, the final project 20%, and the final exam 30% of the final grade.

Honor Code

Students are permitted (and encouraged) to discuss solution approaches to the weekly projects with classmates, however there is to be no wholesale copying of code. For the final project, you may discuss your solution approach with your classmates and others, but you MAY NOT look at each others code. We want to see your individual efforts. In class exams are closed books (and internet), however students are permitted to have one hand written 8.5x11 inch sheet of notes.

Algorithm Assignment 1

due in class January 21

Do the following:

- Before the tutorial, download the tutorial file (matlab program) for tutorial 1. The directory of the tutorials is given here. Run through the tutorial to refresh your memory on how to do simple plotting and matrix manipulations.
- At the tutorial (or before, if you like), get started solving the algorithm problems.
- In class next Thursday (1/21) turn in your solutions!

The Problem:

The first algorithm assignment is given here

CBE 20258 Algorithm Problem Set 1

Due in class 1/21/16

Problem 1). Setting up Systems of Equations in Matrix Form:

a. Suppose you have a specified number of cans of red, blue, and yellow paint. You don't like these colors, however, and want to combine them to form cans of purple, orange, and green paint (which are composed of even mixtures of the appropriate primary colors). If the number of cans of the three primary colors is given by a 3x1 column vector $\underline{b}$ and the number of cans of the new colors is given by a 3x1 column solution vector $\underline{x}$, set up the problem in the form $\underline{\underline{A}}\,\underline{x} = \underline{b}$, clearly identifying $\underline{\underline{A}}$, $\underline{x}$, and $\underline{b}$.

b. If you had 5 cans of Red, 2 cans of Blue, and 6 cans of Yellow, how many of each of the new colors can you make? You can solve this easily by hand, or use the computer if you wish.

c. What is wrong with your solution if you had 8 cans of Yellow?

Problem 2). Algorithm vs. Numerical Error: You are trying to design a system to measure the temperature gradient of a slab of material normal to its surface (e.g., the normal derivative of the temperature evaluated at x = 0). You have one probe at the surface itself (x = 0), and can place a second at a depth h (x = h). The probes only measure the temperature to a precision of ± 0.1°C, the first derivative is expected to be of O(1 °C/cm), and the second derivative should be of O(2 °C/cm^2). About how deep should you place the second probe? What is the expected precision of your measured gradient? Show your work and assumptions!

Problem 3). Overflow and Algorithm Issues:

a. The lottery has reached amazing heights lately, in part due to the decrease in the individual odds of any set of numbers winning from 1/175M to 1/292.2M done last fall (actually a great marketing move...provided they pay out pretty soon!). The drawing on 1/13/16 is expected have a jackpot of $1.3B, and it is expected that $1B in tickets will be sold at $2 each. Calculate the probability that there will be exactly 0, 1, 2, 3, and 4 winners of the jackpot. Assume that everyone uses the "computer option" to pick their numbers at random. Hint: Although you could use the example program from class (lecture 2), it takes a little while to run and is very inefficient for such large numbers! You can do it with a calculator if you recognize that for small k and large n, (n-k)! ~ n!/n^k. The "1-p" exponentials are dealt with using the first term of the Taylor Series: $\ln(1-p) \approx -p$ for small p.

b. Interestingly, research has shown that if people pick their own numbers rather than letting the computer do it, the choices are highly non-random. In particular, people usually pick numbers from 0-31 (corresponding to birthdays) while the lottery numbers go from 0-69. In view of this, *qualitatively* answer the following questions:

1) Does the probability of any individual's lottery ticket winning change?

2) Does the probability of getting multiple winners change (e.g., do you expect the payout to a winning ticket decrease due to splitting the prize)?

3) Does the probability of no one winning the lottery change?

4) For financial reasons, why are you better off letting the computer pick the numbers than picking birthdays?

5) Can you think of an even better strategy which would lead to a higher expected return (it's still a bad investment, though...)?

Problem 4) Ill-Conditioned Equations: A classic example of ill-conditioned equations occurs in acid-base equilibria. Suppose you are diluting 60g of acetic acid (CH_3COOH) with water to make up a 1 liter solution. The goal is to determine the pH (the negative of the log10 of the hydrogen ion concentration [H+]) of the solution. There are four species present: [H+], [OH-], [CH_3COOH], and [CH_3COO^-]. These are governed by the four balance equations:

1) Mass balance: [CH_3COOH] + [CH_3COO^-] = Total Molarity of Acetic Acid

2) Ion balance (electroneutrality): [H+] = [OH-] + [CH_3COO^-]

3) Water Equilibria: [H+]*[OH-] = Kw = 10^{-14} M^2

4) Acid Dissociation: [H+]*[CH_3COO^-]/[CH_3COOH] = Ka = 1.76×10^{-5} M

It is possible to just rearrange these (non-linear) equations so that the right hand side of each is zero, and then solve it as a system of non-linear equations $\underset{\sim}{f}\left(\underset{\sim}{x}\right)=0$. Matlab is very good at finding the roots to systems of equations, and the command we will use later this term to do this is "fsolve". If you do this, however, you will find that you often get weird and incorrect answers, including negative concentrations of some of the ions! This is because the problem as written is ill-conditioned (variables and equations differ widely in magnitude). So:

Recast these equations so that they are well-conditioned (all variables $\underset{\sim}{x}$ and equations $\underset{\sim}{f}$ are of comparable magnitude).

For an extra credit point, solve for the pH using Matlab...

Hint: Try playing around with taking the log of the equilibrium equations and working with the log of the concentrations...

Why Getting Numerical Analysis Right Matters:

Or a Layman's View of Economic Disaster

Definitions

- CDO: Collateralized Debt Obligation - A mixed up collection of mortgages or other debts (designed to reduce risk).
- Tranche: A way of dividing up a CDO to make pieces less (senior) or more (mezzanine) risky.
- CDO^2: A recombination of mezzanine CDO's into new senior and mezzanine tranches.
- CDS: Credit Default Swap - A weird sort of life insurance for bonds and mortgages (designed to reduce risk).
- Copula: The covariance of default risk.

The CDO Tranches

- Major investors such as pension funds can only invest in AAA bonds, as they are risk averse.
- Subprime mortgages are intrinsically risky (that's why they are subprime!).
- To make an investment grade bond, a bunch of subprime mortgages are combined into a pool, diversifying the risk.
- This pool is then subdivided into *tranches*. Any defaults are assigned to the lowest tranche first (potentially wiping it out). The senior tranches are the last to get defaults.
- The bottom (mezzanine) tranches were then pooled together, and then redivided into tranches again, producing the CDO^2 financial product.
- The top tranches were sold off to banks and pension funds, creating a huge market for subprime loans.

The CDS

- Banks like to control risk by insuring against loss, which is an important way of protecting the financial system.
- CDO's were insured via something called a *credit default swap*.
- Insurance was sold that would pay out if the CDO defaulted, and the insurance company got a premium for their risk.
- Rather than just selling the CDS to the bond owner (which makes sense), AIG and other insurers sold CDS's to anyone, including hedge funds which recognized that the insurers were underestimating the risk.
- This is sort of like taking out a life insurance policy on your neighbor, without even letting him know…
- The total exposure on CDS's was actually far greater than the total CDO market.
- Goldman and deal makers made fees on every transaction.

Risk Estimation

- In order to price the insurance and rate the risk of the CDO’s, it was necessary to determine the probability of default.
- Because they were pooled investments, the default rate was critically dependent on the covariance of the subprime mortgages.
- Rather than examining the bonds making up the CDO, David Li proposed that the market price of the related CDS could be used to determine the *copula*.
- This worked great, except all the available data was during a period of rising home prices - when the covariance was naturally small.
- When the housing market started to go down, really bad things happened…

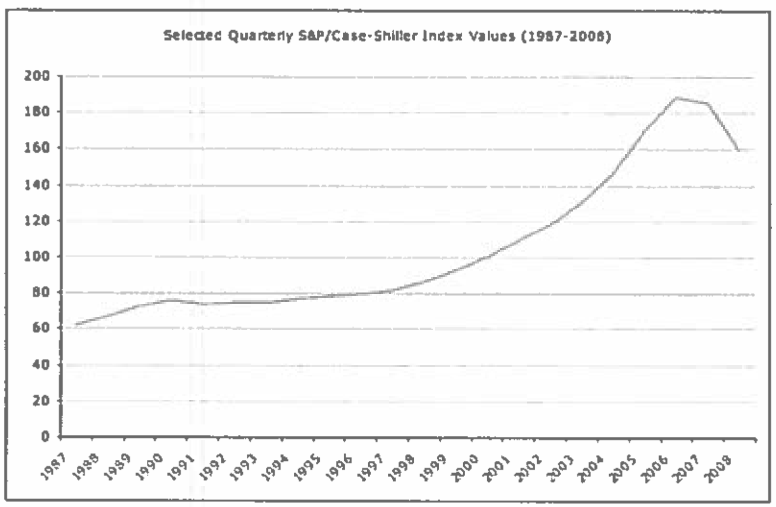
The Effect of Greed!

- Wall Street was rewarded via fees for brokering deals.
- Because a market was created for sub-prime (read "bad") loans, lenders solicited wildly inappropriate borrowing and passed the bad loans off to others. This increased the total amount of borrowing far beyond a sustainable level.
- Insurers would issue CDS guarantees in an amount far greater than the CDO being insured: again, they would get lots of fees.
- Provided that housing prices kept going up, the banks and the insurers thought they had a great deal!

Why the error?

- CDO correlation used CDS valuation, only available during a period of rising home prices!
- Covariance is massively underestimated in an adverse housing market.
- Reality was too painful to believe, and easier to ignore.
- Senior management used the "black box" calculations and didn't really understand the implications of the underlying assumptions.

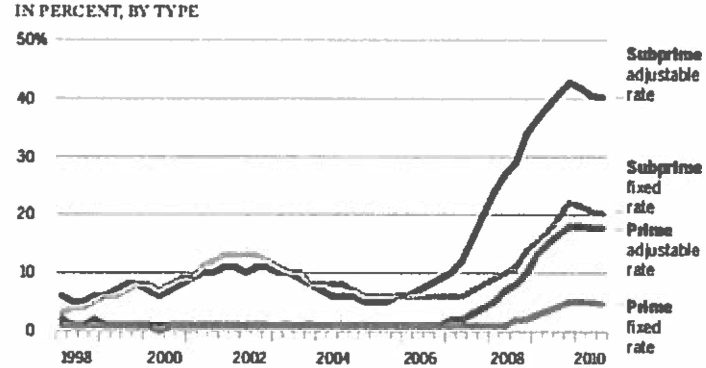
Housing Price Variation



Mortgage Delinquencies by Loan Type

Serious delinquencies started earlier and were substantially higher among subprime adjustable-rate loans, compared with other loan types.

IN PERCENT, BY TYPE



NOTE: Serious delinquencies include mortgages 90 days or more past due and those in foreclosure.
SOURCE: Mortgage Bankers Association National Delinquency Survey

Leverage: Compounding the error

- Leverage means borrowing money to invest in a higher yielding opportunity
- Leverage can be very good, or very very bad.
- You have to accurately asses the risk of the investments
- Financial institutions underestimated the risk of CDO's, so what they *thought* was a sure thing, was really a sure disaster
- At the end, ML was leveraged 30:1 in CDO's, with predictable results…

A Timeline

- 1997 housing price index = 80
- 1997 CDS invented by JP Morgan Chase
- 2000 Paper by David Li "On Default Correlation: A Copula Function Approach" - determined default correlation from CDS pricing
- 2000 "Commodity Futures Modernization Act of 2000"- A "late night" bill deregulating credit default swaps
- 2001 $920 billion in CDS outstanding, $275 billion in CDO's
- 2006 peak housing price index of 190 (2.4x in 9 years!)
- 2006 CDO market hits $4.7 trillion (17x increase in 5 years!)
- 2007 CDS market hits $62 trillion (67x increase in 6 years!)
- 2008 Lehman Brothers declares bankruptcy
- 2009 Troubled Asset Relief Program costs the taxpayer nearly $1T

The Take Home Message:

- Numerical & Statistical Analysis can be very useful to understand complex systems!

- You must -always- question the assumptions that go into any numerical model, particularly one involving evaluation of risk/uncertainty.

- Most systems and models will 1) assume some level of independence, and 2) underestimate the degree of covariance.

- DON'T BELIEVE THE ANSWER JUST BECAUSE THE COMPUTER GIVES YOU A NICE GRAPH OR NUMBER! UNDERSTAND WHAT YOU ARE DOING!

Further Reading

- Michael Lewis, "The Big Short", 2010.
- Felix Salmon, "Recipe for Disaster: The Formula That Killed Wall Street" Wired.com 2009.
- Anna Barnett-Hart, "The Story of the CDO Market Meltdown: An Empirical Analysis" Senior Thesis, Harvard University 2009.

Review of Matrix Ops:

A matrix is a way of representing a large amount of information in a compact way.

Matlab ≡ Matrix Laboratory

⇒ it's all about manipulating matrices!

Suppose we look at the HW grades for a class. Let:

G_{ij} ≡ grade of the i^{th} student on the j^{th} homework

So the matrix $\underset{\approx}{G}$ might be:

$$\underset{\approx}{G} = \begin{bmatrix} 9 & 8 & 7 & 8 \\ 10 & 6 & 9 & 10 \\ 2 & 3 & 1 & 0 \end{bmatrix} \begin{matrix} \leftarrow 1^{st} \text{ student} \\ \leftarrow 2^{nd} \text{ student} \\ \ \end{matrix}$$

↑ 1^{st} assignment ↑ 2^{nd} assignment

Suppose we want to calc. the <u>average</u> for each student.

Let $\underset{\approx}{G}$ be an $n \times m$ matrix (n rows, m columns)

Then the <u>vector</u> $\underset{\sim}{AG}$ is:

$$\underset{\sim}{AG} = \underset{\approx}{G} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{m} \equiv n \times 1 \text{ vector!}$$

$n \times m$ matrix ⟷ $m \times 1$ vector

these <u>must</u> match!

We multiply each <u>row</u> of <u>first</u> matrix by each <u>column</u> of 2<u>nd</u> matrix (only one here)

So $\underset{\approx}{A}\,\underset{\approx}{B} = \sum_j A_{ij} B_{jk} = C_{ik}$

which is a series of nested loops!

In Matlab, this is a lot more compact!

$\underset{\sim}{AG} = \underset{\approx}{G} * \text{ones}(4,1) * \frac{1}{4}$

$\underset{\sim}{AG} = \text{sum}(\underset{\approx}{G}')' * \frac{1}{4}$

(sum command sums over columns! we use ' to take transpose!)

$\underset{\sim}{AG} = \text{sum}(\underset{\approx}{G}, 2) \frac{1}{4}$

↑ forces sum over rows!

All yield the same result!

Final note: Matlab scripts & functions

Run example!

- Break your code into functions and scripts

- comment your code so you can remember it!

- define function I/O & purpose in comment on function (1st line)