CLASS DESCRIPTIONS—WEEK 3, MATHCAMP 2013

Contents

9:10 AM Classes

Multilinear Algebra. (🌶🌶🌶🌶, Asi & Waffle, Wed–Sat)
Multilinear algebra is a theory that gives us ways to deal with operations that are linear in each argument. The beginning of the story is *bilinear forms*, which are functions of two vectors that are linear in each variable (which is to say, they behave like "multiplication"). Wouldn't it be nice if we had a *linear* function that captured the same information? But how can we come up with a function of just one argument that achieves this goal? This question leads to the construction of the tensor product, which can be thought of as a way of "multiplying" vectors together.

In this class, we will develop more machinery from multilinear algebra. For example, the tensor product is in some sense the "habitat" of multilinear forms. Similarly we will create "habitats" for multilinear *symmetric* and *anti-symmetric* functions of vector spaces. As a consequence, we will get beautiful, conceptual, basis-free definitions of functions such as the determinant, as well as other interpretations of algebraic objects such as polynomial rings.

Homework: Recommended

Prerequisites: Linear algebra and ring theory.

Quantum Computing. (🌶🌶🌶, Matt, Wed–Sat)
Quantum computers, instead of acting on just ones and zeros, can act on quantum superpositions of them: we can have a "50% one, 50% zero," for example. We'll see some surprising things that this extra power gives us, what a quantum computer program could look like, and finally see Shor's algorithm for quantum factoring.

Homework: Optional

Prerequisites: Familiarity with matrices and complex numbers.

Surfaces and Symmetries (week 2 of 2). (🌶🌶, Susan, Wed–Sat)
Week 2 of Surfaces and Symmetries! If you've taken week one, this is for you!

Homework: Recommended

Prerequisites: None

10:10 AM Classes

Harmonic Analysis. (🌶🌶🌶🌶, Ari & Dave, Tue–Fri)
One recurring theme in mathematics is breaking large, complicated things into small, simple pieces that you know how to handle. For example, you'll often write vectors in an orthonormal basis to make calculations easier. But how do you find a good basis for, say, continuous functions? In this class, we'll develop the Fourier transform, a powerful tool for approximating sufficiently well-behaved functions on the unit circle, $\mathbb{R}$, or any other locally compact abelian group. For instance, we'll see that once we've done the hard work of developing the right tools, it is easy to use this machinery to prove the isoperimetric inequality: a circle encloses greater area than any other shape with the same perimeter. We'll also explain how to prove some amazing identities, such as

$$\sum_{n=-\infty}^{\infty} \frac{1}{(n+\alpha)^2} = \frac{\pi^2}{(\sin \pi\alpha)^2}.$$

Homework: Recommended

Prerequisites: Linear Algebra, some amount of Real Analysis, Topology, and Abstract Algebra

Related to (but not required for): Elections: Influence and Stability (week 4)

Representation Theory of Finite Groups (Week 1 of 2)
(Week 1: 🌶🌶🌶 , Week 2: 🌶🌶🌶🌶, Mark, Tue–Fri)
It turns out that you can learn a lot about a group by studying homomorphisms from it to groups of linear transformations (if you prefer, groups of matrices). Such a homomorphism is called a *representation* of the group; representations of groups have been used widely in areas ranging from quantum chemistry and particle physics to the famous classification of all finite simple groups. For example, Burnside, who was one of the pioneers in this area along with Frobenius and Schur, used representation theory to show that the order of any finite simple group that is not cyclic must have at least three distinct prime factors. (The smallest example of such a group, the alternating group A_5, is important in understanding the unsolvability of quintic equations by radicals.) We may not get that far, but you'll definitely see some unexpected, beautiful, and important facts about finite groups in this class, along with proofs of most or all of them. The first week of the class should get you to the point of understanding character tables, which are relatively small, square tables of numbers that encode "all" the information about the representations of particular finite groups; these results are quite elegant and very worthwhile, even if you go no further. In the second week, the chili level will ramp up a bit as we start introducing techniques from elsewhere in algebra (such as algebraic integers, tensor products, and perhaps modules) to get more sophisticated information.

Homework: Recommended

Prerequisites: Group theory and linear algebra. If you are uncomfortable with abstraction, you will probably not enjoy this class.

Related to (but not required for): Representations of the Symmetric Group, Multilinear Algebra

The Graph Minor Theorem. (🌶🌶→🌶🌶🌶, Alex & Marisa, Tue–Fri)
Some proofs take a paragraph. The proof of the Robertson-Seymour Theorem took 500 pages across 20 journal articles, published between 1983 and 2004. We will... not prove it this week.

Instead, we will tell you why graph theorists consider it one of the greatest results in the history of combinatorics. (High bar!) Simply put, the theorem tells us that in any infinite family of graphs, some graph is a minor of another graph. This innocent-sounding statement yields powerful results, such as the existence of efficient algorithms to test whether we can draw a graph on a given surface.

You might have seen a theorem that says that a graph is planar iff it contains no K_5 or $K_{3,3}$ minor. But what if, instead of drawing our graph on the plane, we wanted to draw it on some arbitrary surface S: do we have an analogous set of "forbidden" minors? Is that set finite? These are questions we'll ask and answer using the Robertson-Seymour Theorem. We'll also discuss some of the ideas in Robertson and Seymour's proof (as much as we can fit in a 4-day, less-than-six-chili class).

Homework: Optional

Prerequisites: Intro to Graph Theory (or equivalent)

Related to (but not required for): Graph Theory, Posets, Surfaces and Symmetries

11:10 AM Classes

Continued Fractions. (🌶🌶, Dave, Tue–Fri)
Problem #4 from the Mathcamp 2010 qualifying quiz asked: "If the integers 2^n and 5^n begin with the same ten digits, what are those digits?" You might wonder how to show that such an n exists; or, better yet, how to find the smallest value of n with this property. (The order of magnitude of the smallest n is around a billion, so an exhaustive search won't do you much good!) It turns out that this is linked to finding good rational approximations to $\log_{10}(2)$. To understand this sort of approximation, we will develop the theory of *continued fractions*, which are expressions of the form

$$3 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cdots}}}.$$

As a bonus, we'll also figure out how to find the smallest integer solutions (x, y) to the equation $x^2 - ny^2 = 1$ for any n.

Homework: Recommended

Prerequisites: Know what a recursive sequence is; some comfort with the notion of a limit

Polynomial Method. (🌶🌶🌶🌶, Pesto, Tue–Fri)
Given n points in $\mathbb{R}^2$, there's a nonzero two-variable polynomial of degree at most $2\sqrt{n}$ that's 0 at all of them, since the dimension of the vector space of such polynomials is more than n. This technique and the existence of such low-degree polynomials are useful for proving a variety of things, like:

(1) In SET generalized to have p values per characteristic (instead of 3), if you have at most $.0002p^4$ cards, then you can add a card without adding any sets.
(2) If almost half the values of a polynomial get changed by a malicious opponent, we can reconstruct the polynomial.
(3) Bezout's Theorem: if P and Q are (two-variable) polynomials with no common factor, then their number of common zeros is at most the product of their degrees.

Homework: Recommended

Prerequisites: Linear algebra, enough to understand the first sentence of the blurb. (If you're not sure, talk to Pesto.)

Symmetric Polynomials and Schubert Calculus (week 2 of 2). (🌶🌶🌶, Kevin & Nic, Tue–Fri)
This is a continuation of the first week of Symmetric Functions and Schubert Calculus. The first week is not required—talk to us if you are interested in jumping in!

Homework: Recommended

Prerequisites: Linear Algebra

1:10 PM Classes

Functions of a Complex Variable (week 2 of 2). (, Mark, Tue–Fri)
This is a continuation of last week's class. For more information, see last week's blurb and/or talk to Mark.

Homework: Recommended

Prerequisites: Week 1 of this class.

The Traveling Salesman and Friends. (, *Gwen Spencer*, Tue–Fri)
Given a set of cities, and travel times between them, find the shortest tour that visits every city. Though it may sound like a simple task, this is the famous and (infamous!) Traveling Salesman Problem (TSP). This course will develop some basic ideas about algorithmic thinking, focusing on this notorious optimization challenge. For a simpler friend of the TSP, we'll see that being greedy leads us directly to the optimal solution. With the Traveling Salesman Problem, things are not so simple: being greedy may result in some decisions we wish we could take back. How badly can greedy be tricked? Good algorithms can never be tricked, and will produce a answer in our lifetime. We'll finish with a gift from the Traveling Salesman's simple friend: a pretty good algorithm.

Homework: Required

Prerequisites: Basic Graph Theory

Voting Theory. (, Alfonso, Tue–Fri)
When a large group of people have to make a decision together, bad things can happen. For example, suppose that a group of 10 campers is trying to decide which game to play tonight. Suppose further that 3 of them want to play dominion, and the remaining 7 would prefer to play *any* game they can possibly think of other than dominion. If the remaining 7 are divided between 5 or 6 different games, a strict plurality election system will force them to play dominion, even though a majority of the 10 campers would be thoroughly unsatisfied. It seems, then, that the plurality election system is unfair. What could we do to make it fair?

Which election system are the most fair? What does *"fair"* mean, anyway? How does a mathematician try to solve this problem? Come to this class and find out. *Warning:* Your faith in democracy may vanish after this class.

Homework: Optional

Prerequisites: None

Earlier Superclasses: 9:10am-11:00am, 3:10pm-4:10pm

Communication Complexity. (, *Marcin Kotowski*, Tue–Fri)
Alice and Bob watch 16 teams play in a tournament. Alice knows who played in the finals, but forgot who won. Bob knows who won, but forgot who the opponent was. How many bits of information do they need to communicate so that Alice knows who won? (Can they do it with less than the obvious 4 bits, if all they know are the names of all teams and the information listed above?)

In the course, we will study problems of this kind: how much communication is needed to compute a given function, if two parties only get parts of the input? Such questions provide a nice, concrete model of computation in which it is possible to prove lower bounds (as opposed to the usual complexity theory)—for example, we will prove that $P \neq NP$![1] We will see a variety of communication models and lower bound techniques, with applications to data structures and Boolean circuits. If you want

[1] In the communication complexity model

to see Alice and Bob communicate (provably optimally!) their innermost, intangible thoughts, this course is for you!

Homework: Optional

Prerequisites: Basic probability (random variables, independence, expectation value). Knowledge of linear algebra is welcome, but not required.

Related to (but not required for): The Forehead Game, Simple Models of Computation

Machine Learning. (🌶🌶, Mira & Tim!, Tue–Fri)
Machine learning is the construction and study of computer systems that can figure out ("learn") patterns from data without being given an explicit description of the patterns they are looking for. In other words, it is an attempt to make machines do things that humans do effortlessly (understand speech, recognize faces, learn new concepts)—but to do it with huge data sets that are too big for humans to deal with directly.

For instance, suppose we want to train a computer to distinguish email spam from non-spam. Humans are generally pretty good at this task, but would have a hard time explaining to a computer exactly how they know that something is spam: usually, it's just "obvious" to them. Instead, we want a general-purpose technique that would allow us to give the computer a bunch of messages labeled "spam" and "not spam" and have the computer itself figure out what distinguishes the two categories, so that if we give it a message it has never seen before, it has a good chance of classifying it correctly.

In this example, we had to give the computer some "training data" (the labeled messages) because we wanted it learn a *particular* pattern that we care about (spam vs non-spam). This is called **supervised learning**. But sometimes, we ourselves don't know exactly what pattern we are looking for. For instance, suppose we have Mathcamper votes for proposed Week 5 classes, and we want to find some structure in this data that helps Nina come up with a good Week 5 schedule. In this case, there is no training set of right and wrong answers: we just want the computer to look for patterns of various kinds. This is called **unsupervised learning**, and the algorithms it uses are quite different.

In this class, we'll discuss some of the most basic techniques of both supervised and unsupervised learning (categorization and clustering respectively). In-class assignments will include both paper-and-pencil math problems (there's a lot of cool math behind machine learning!) and the chance to play with some of the algorithms on the computer. No programming experience is required, since you will not be doing any serious coding during the class (it would take too long). However, if you do have some programming experience, we'd be happy to supervise a project based on this class where you would actually get to do some programming.

Homework: None

Prerequisites: Calculus (through integration) and probability (solid understanding of random variables, mean, and standard deviation; week 1 of Aaron's course is sufficient). No programming experience assumed.

Related to (but not required for): Extreme Probability; distantly related to other Mathcamp CS classes.

Mathematical Sculpture. (🌶, *George Hart* & Zach, Tue–Fri)
Come transform ordinary items into extraordinary geometric sculptures! In these large-scale, collaborative construction projects led by Zach and the acclaimed mathematical sculptor/expositor George Hart, we will work together to assemble intricate, puzzling polyhedral jumbles, room-filling rubber-band knot-webs, and much more. Assembling these creations requires scrutiny of their beautiful mathematical underpinnings from such areas as geometry, topology, and knot theory, so come prepared to learn, think, and build! Browse `http://www.georgehart.com/sculpture/sculpture.html` and `http://zacharyabel.com/sculpture/` for examples of these artists' work.

You can drop into individual days of this class without taking previous days.

Tue 3D Zometool workshop. Come explore a range of polyhedral and other structures in a hands-on construction workshop. Along the way, we'll discover several interesting patterns and relationships concerning 3D geometry and symmetry.

Wed Giant 4D polytope construction with Zometool. Polytopes in four dimensions can be projected down to 3D as a tool for visualizing their structure. Come make a large model of a uniform polytope from the 120-cell family and learn to stretch your 4D visualization muscles.

Thu Humongous polyhedral rubber band web. We'll use thousands of rubber bands to assemble a giant, bouncy mesh, and along the way, we'll explore its polyhedral symmetries and the nontrivial knot theory holding it together.

Fri Binder clip monstrosities. Binder clips are a surprisingly versatile medium, and their inevitable limitations lead to fun graph-theoretic considerations. We'll work together on an intricate binder clip monolith, and you'll also make some smaller sculptures to take home.

Homework: None

Prerequisites: None

Later Superclasses: 11:10am-12:00pm, 1:10pm-3:00pm

Combinatorics of Partitions. (🌶🌶, *Michał Kotowski*, Tue–Fri)
A partition of a natural number N is a representation of N as a sum of positive integers (disregarding their order). For example, $5 = 4 + 1 = 3 + 2 = 3 + 1 + 1$ and so on. In how many ways can one write N as sum of odd terms only? Or what if we require that all terms be distinct? It turns out that the number of ways to do this is exactly the same in both cases. This is the simplest type of a "partition identity" that we will study in this course. This may look like magic, but much weirder identities are true. For example, the number of partitions with no consecutive parts (i.e. no parts like 3 and 4 or 5 and 6) is the same as the number of partitions with parts congruent to 1 or 4 modulo 5 (this is one of the so-called Rogers-Ramanujan identities).

The main focus of this class will be on bijective proofs—showing that there is exactly the same number of partitions of one type and of another type by providing an explicit pairing of partitions from one group and the other group. Such proofs often require clever ideas and may be said to belong to the "heart" of combinatorics. Along the way we will draw a lot of pictures—so-called Young diagrams of partitions.

The other approach to partitions will be through generating functions, a certain kind of formal power series. By manipulating them, one can magically obtain easy proofs of many combinatorial identities without going through explicit bijections at all. In fact, purely bijective proofs of many identities are not known to this day, despite efforts of great mathematicians such as Ramanujan! We will taste both the power and the limitations of generating functions and see the interplay between pure combinatorics and algebra.

Homework: Optional

Prerequisites: Familiarity with bijections, binomial coefficients and geometric series, like $\frac{1}{1-x} = 1 + x + x^2 + \ldots$, would be welcome.

Commutative Algebra. (🌶🌶🌶, Ruthi, Tue–Fri)
Do you love algebra? Have you heard about rings and their properties and want to know lots more? Have you always felt that polynomial rings are one of the nicest rings you could imagine?

Commutative algebra is the language of both algebraic geometry and algebraic number theory, and thus is essential for the study of these fields (pun intended). In this class, all rings will have a 1 and multiplication will always be commutative. But then things start getting hard!

The flavor of this class will be very different than the standard class. You're going to be doing lots of talking to each other, lots of working on problems and very little listening to me tell you things. Instead, you'll be thinking about the theorems and proving conjectures full-time!

Homework: Optional

Prerequisites: Intro to Ring Theory

Related to (but not required for): Linear algebra, multilinear algebra, flag varieties, representation theory, algebraic number theory

(Th)ink Machine. (🌶, Aaron, Tue–Fri)
Computation is the process of sculpting with information—cutting it down, drawing it out, and shaping it into something new. Like other forms of sculpture, it's both fun to do and beautiful to watch. It's also a creative activity: its results can surprise us, and tell us things we hadn't known before.

The aim of this course is to introduce the basics of computer science in a way that emphasizes the aspects described above. We'll see how the idea of "sculpting with information" can be turned into a rigorous definition of computation, and then we'll use our definition to explore the limits of what computers can do.

If you love building machines, making art, solving puzzles, or proving by example, this is the course for you. You'll have a lot of creative freedom, and a lot of options for what you want to study. Whether you're a total computer science beginner or a seasoned Turing machine wrangler, you should be able to find something fun to do.

Homework: None.

Prerequisites: None, but there will be lots of fun for students who took Toy Models of Computation in Week 2!

Colloquia

Spread in Networks: Containing the Harmful, Encouraging the Virtuous. (*Gwen Spencer*, Tue)

Math is Cool! (*George Hart*, Wed)
With his geometric sculptures, puzzles, videos, and hands-on workshop activities, George Hart actively demonstrates that math is a fun, creative subject—i.e., that Math is Cool! He also is a co-founder of the Museum of Mathematics that recently opened in New York City. Come see him present and discuss a range of topics from his creative output. For examples of his work, see `http://georgehart.com`

Presentations of Qualifying Quiz Solutions. (The Staff, Thu)

Simulation as the engine of common sense. (*Josh Tenenbaum*, Fri)

Visitor Bios

George Hart loves to take cool geometric ideas and translate them into tangible physical form, for example by making geometric sculpture. He is a co-founder of the Museum of Mathematics in NY, where he was Chief of Content during its design phase, but is now a freelance mathematician. When he visits Mathcamp, he likes to try out new ideas for hands-on workshops.

Gwen Spencer (Dartmouth/Smith) specializes in combinatorial optimization. Functions are defined over patterns of many light switches: each switch is either "on" or "off." The total pattern of switches determines the value of the function. Such functions encode many classic and contemporary optimization problems in networks including applied problems in ecology, but they are tricky to optimize since derivatives don't exist. Further, while a solution to a particular problem is nice, what we really want is a method that solves every single example of a type of problem. How can optimal (or provably-near optimal) solutions be found?

Marcin Kotowski is a PhD student in Toronto. He works at the interface of probability and geometric group theory, studying the interplay between geometric and algebraic properties of groups. This includes concepts like random groups and stochastic processes on graphs and groups. He is also interested in many non-academic things: books, music, mountains, Japanese mahjong, improv theater, surreal humor and dreams (which often incorporate math, synesthesia and nihilism).

Michał Kotowski is currently a graduate math student in Toronto. He did his previous degree in Warsaw, Poland. His main interests are geometric group theory and probability, especially the interplay between these two fields. This includes questions like: how does a group look like from "far away"? How quick is a random walk on an infinite graph or group? Can one hear the shape of a random graph? Apart from math, Michał is interested in physics and computer science, and apart from science—in lots of stuff, including music, literature, dreams, mountain hiking, video games and teaching gifted kids.

Josh Tenebaum (MIT) is a professor of cognitive science and a member of the MIT Computer Science and Artificial Intelligence Lab (CSAIL). In his research, he builds mathematical models of human and machine learning, reasoning, and perception. His interests also include neural networks, information theory, and statistical inference.