EECS 70 Discrete Mathematics and Probability Theory
Summer 2020 Note 0

A perspective on EECS as a whole and 70's role within it

In this introductory mini-lecture, we will try to convey the big picture at a "cultural" level. 70 is a basic second-year lower-division course. It is one of the gateways[1] into the EECS department's upper-division courses, following the first-year 16AB series that focus on modeling, continuous mathematics, and its connections with the physical world and machine learning. This 16A, 16B, 70 trilogy is in parallel to the more programming-oriented 61ABC. The only stated prerequisite for 70 is "sophomore mathematical maturity" — which can seem confusing. One easy way to understand this is that it just means that you have reached the mathematical maturity level that we expect students to have reached by the end of the 16AB sequence if they are sincere, diligent, and are understanding what is going on reasonably well. But what does this mean more concretely? The challenge with a word like "maturity" is that it is hard to exactly enumerate the combination of attributes this is referring to. But a decent approximation is:

- Being able to work systematically on a mathematical problem without being overwhelmed or paralyzed by fear.
- Having the emotional and mental endurance to do longer derivations or longer calculations in a step-by-step manner, being able to check your own work both as you go and at the end. Being able to maintain sustained focus in support of this.
- Not needing to follow an exact recipe or to simply pattern-match (at a surface level) to a similar solved problem to be able to start working — instead being able to systematically write down what it is that you know, where it is that you want to get to, and assigning variables and notation as appropriate to the relevant objects/quantities in the problem. Being able to get started doing this even when things are unfamiliar — modeling problems.
- A solid working knowledge of calculus (both differential and integral) and basic linear algebra, as well as both the modeling and problem solving strategies associated with them. For example, having internalized the experience of doing integrals and having to deploy different strategies to massage intermediate results into forms that you can work with.
- Being able to keep track of the difference between what you know, what you're trying to get, and the intermediate results that you have already established on your journey. Some sense of the natural flow towards simplicity and what you want.
- Understanding definitions, being able to parse mathematical expressions, and being able to express desired things mathematically.
- Experience reasoning about numerical procedures like Gaussian Elimination and understanding why they work and what the steps represent.

[1] Regardless of which door you are coming through: College of Engineering or Letters and Science. The door into campus is irrelevant when it comes to what's actually important.

- Exposure to and experience following (being led through the steps) straightforward proofs in direct form, contradiction form, and induction form.
- Some appreciation for how when facing an unknown problem, it is useful to think about and work through simple examples or zoom in on simpler cases.
- Being able and willing to try an approach without knowing for sure that it will succeed, taking steps, and then backtracking when it doesn't, and being willing to try something else instead of giving up.

As you can see, much of the above is fundamentally emotional in nature, which is not surprising with a word like "maturity." Without this basic maturity (which like most maturity, is a function of experience), it will be very hard to get the most out of 70 and do well. That said, your hard work in 70 will itself strengthen your mathematical maturity significantly further. In addition, if you approach the course correctly, small deficiencies in your prior maturity can get filled as you go.

Anyway, as you might have heard, this is a pretty mathematical course. The official title is "Discrete Mathematics and Probability Theory" and the course catalog description describes the content as:

> Logic, infinity, and induction; applications include undecidability and the stable marriage problem. Modular arithmetic and GCDs; applications include primality testing[2] and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing, existence arguments, Bayesian inference.

That might look like a grab bag of esoteric math stuff, especially when you contrast it with the more clearly useful continuous math topics you've seen in 16AB which are largely building machine learning foundations[3] with mathematical topics being discovered/developed in the context of explicitly practical problems like imaging/comp-photo, touchscreens, GPS, and making cyborgs. You might be wondering, why is there such an overtly mathematical course sitting right at the threshold of the majors (or minors) offered by the EECS department? After all, isn't EECS a collection of practical fields? Isn't it about making computers, designing cell-phones and wireless networks, programming applications, creating robots, making search engines, creating computer graphics, and enabling all sorts of advanced technology?

What does a bunch of math have to do with anything? Especially things like proofs. Many outsiders think that what practical engineers and designers need are ways to do calculations, while the proofs are something arcane that are best left to the mathematicians who care about these sort of things. The reality is quite different, as you've already tasted in the 16 series. To paraphrase Prof. Bob Gallager[4], "in Engineering problems, the theorems seldom apply directly, but the proofs often do." Some of you may think that Mathematical rigor is stifling. In reality, mastering the rigor of proofs *liberates* your creativity — you are free to follow

[2] We're actually going to skip primality testing this term.

[3] 16A Math topics: Linear inverse problems, systems of linear equations, and how that motivates basic linear algebra and measuring information in degrees of freedom (the n equations for n unknowns barrier); Gaussian elimination and why it works; eigenvalues and their connection to web graphs and PageRank; inner-product structure, projections, and least-squares for approximately solving overdetermined systems of equations; discrete-inference by maximum correlation, and orthogonal matching pursuit (OMP) for finding sparse solutions to underdetermined systems of equations — breaking the n equations for n unknowns barrier.

16B Math topics: Solving systems of linear differential equations and linear recurrence relations; changing coordinates using eigenbases; stability and dynamic behavior; learning system models from data; Gram-Schmit orthonormalization and its value in speeding up algorithms; recursive upper-triangularization and the spectral theorem; the singular value decomposition (SVD) and low-rank approximation (PCA), minimum norm solutions to underdetermined systems of equations; linearizing nonlinear systems and basic vector calculus; polynomial interpolation and approximation, and the Discrete Fourier Transform.

[4] A now retired Prof. at MIT, who has made foundational contributions to information theory, the theory of distributed computation, and computer networks.

flights of fancy and speculation, secure in the knowledge that eventually, rigor will catch you and tell you if you're making a mistake.[5]

In this lecture note, we will try to metaphorically describe how mathematics relates to the EECS department. Although math is important for all science and engineering disciplines, the relationship is particularly close and intense when it comes to EECS. When outsiders normally think of the power and utility of applied math, their intuitive sense is for mathematics as a descriptive and predictive tool. Something that folks can use to help you understand the real world.

Within EECS, that dimension is certainly present. But there is another one too.

Math as magic or wizardry

To understand the intimate relationship that EECS has with mathematics, the playful metaphor of magic is useful. As human beings who are a part of a culture in which "magic" plays an important role in literature, mythology, video-games, movies, etc. you have an intuitive appreciation for magic as an idea — even if we understand that magic is not really a part of our natural world, it is definitely a part of our range of mental models.

What makes magic different from mere understanding is its active force. In stories, a mage wishes not only to understand magic, but to manifest it in a controlled manner. To make magic flow through objects and animate them. That's how folks in EECS intuitively regard math.

Physical electronics

In devices and physical electronics, what folks in EECS do is look for physical materials that "resonate" or respond with certain mathematical properties, and then to weave the materials together into basic devices that are predictably responsive to mathematics. The objects are physical, but they respond in mathematical ways that are particularly interesting. You got a taste for this in 16A when you saw how resistors and capacitors link the physical and electrical worlds mathematically.

Certain key mathematical operations are desired (e.g. gain (amplification), switching, integration, and memory, among others.) and we strive to realize these in ever more efficient ways. But at the same time, folks hunt to discover novel devices that exhibit mathematically interesting or powerful behaviors that might be different. For example, people try to build micromechanical resonators or devices that better harness effects related to quantum superposition[6]. This approach is also being applied to biological systems — rather than simply asking how biology works, EECS people engaged in synthetic biology strive to discover the mathematical potential of biological activity.

All this requires an eye and a *taste* for what sort of mathematical behaviors are interesting, how to recognize them when you see them, and how to enhance them once you do. To use a magical analogy, the discovery of novel devices is like the discovery of new magical herbs, magical ingredients, magical sounds, and magical strokes. But the true power of these magical ingredients is realized in their combination and interconnection. Herbs and ingredients have to be mixed together into potions, magical sounds have to be combined into magical words, and magical strokes have to be put together into magical characters.

[5] Rigor (together with the ability to simulate) lets you "fail fast, fail often, and fail cheap" — and this is one of the secrets to letting creativity lead you to both innovation and deeper understanding.

[6] This is an important part of the quest towards quantum computation.

Circuits

Magic potions, magical words, and magical characters: this is the metaphoric level that is represented by circuits. Devices have be combined and interconnected in a way that lets the math better course through them. Circuits create pathways for mathematical computations. Knowing what can and cannot be harnessed in combination helps one see what one is looking for.

Modern EECS systems involve circuits consisting of billions of discrete entities. But these are built out of smaller groupings in a hierarchical manner, where math is both used to understand how to analyze the interconnections as well as to inspire what the goals of the interconnection itself should be. Once woven together, imperfect devices bring forth more ideal and complex mathematical behaviors.

Modern electronic devices and circuits are truly revolutionary. But within this tremendous general advance, there are three particularly immense capabilities that modern devices and circuits have enabled:

- Automated reliable communication over long distances of vast amounts of information.
- Automated reliable and modifiable storage of vast amounts of information over time.
- Automated programmable universal computation at high speed.

The three of these ongoing triumphs of EECS combine to make our current age possible. Each one has an intuitively magical quality to it that is directly perceptible even to an untrained person.

Programming universal computers

Yet even among the three capabilities above, the third (high speed universal computing) resonates most strongly with magical connotations for human beings simply because we use language — and the realization of universal programmable machines makes actually real what had only been mythical before: magical artifacts that are animated by incantations, incantations that must obey the strict rules of their language.

For people in EECS, the Mathematical Realm is at least as real as it is for most mathematicians. As the famous English mathematician Hardy said: "A mathematician, like a painter or a poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with *ideas.*" But in a sense, for people in EECS, the Mathematical Realm is *even more real* because its inhabitants can actually be made to visit us — subtle mathematical patterns can become physically manifest and useful.

Every computer program is a spell or incantation of sorts. These incantations, whatever the surface form that their language might seem to take, are at their heart mathematical in the source of their power. But what kind of incantations are these? In this metaphor, a universal computer is a kind of magical vessel. And with the right incantations, we actually compel mathematical beings to be summoned forth and animate the vessel. Programming is a kind of conjuring.

And unlike a pure mathematician's math that can surprise only people who can understand it, the math that we in EECS summon forth can surprise even those that can't understand it.

You can't understand our subject without understanding discrete math. While the topics that we are going to cover might seem esoteric to you right now, we will show you examples of their power in the course. Logic is directly related to digital circuit design, and proofs by induction play a central role in analyzing programs. Indeed, induction is deeply connected to the recursion ideas you've seen before in 61A. Consequently, developing your ability to write coherent, rigorous proofs will be an important focus. We will have a unit on a special type of arithmetic known as modular arithmetic. The beautiful properties of such numbers leads

directly to schemes for computer security, and, when combined with the concept of polynomials, for storing and transmitting data so that it is immune to noise.

Programming matter

For us in the EECS department, the idea of programming extends beyond universal computers. Being able to express incantations and abstractions using language is so powerful that it is used even for circuits and the like. Mathematics here is not just the language used to express the desired behavior, it is also used to automatically transform one representation of the behavior into others that are more compatible with the mathematical nature of the physical substrate.

A longer-term view

The centrality of mathematics to EECS means that there is no way that a single semester course like 70 can possibly complete the story for you, even when combined with two semesters of continuous math in 16AB. Instead, our primary goal is to give you more of a taste of the field, with a particular emphasis on showing you both the elegance and power of mathematical thinking. Discrete math lets you see how mathematical thinking has a power that extends beyond the continuous quantities that you have much more familiarity with. There is an aesthetic dimension here (one of the most important things about Math is that it is beautiful, elegant, and its different branches weave together in surprising ways.) and we want you to begin to "feel" what math can do and what powerful ideas are like. To do this, we are going to have to build some foundations and tools, but these will really be fleshed out and elaborated on in subsequent courses.