CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Tuesday, January 9 - Lecture 1.1

Reading: front matter, Chapter 1 and Subsections 2.1.1-2.1.7 of Nielsen and Chuang.

Agenda:

1. Briefly motivating the study of quantum computation from a few perspectives
2. Hoity-toity mathematician perspective: quantum mechanics is a mathematical theory of "quantum probability" or "probability with complex numbers."
3. Linear algebra review: Hilbert spaces, Hermitian/self-adjoint operators, unitary operators, normal operators, spectral theorem (finite-dimensional)

1 Why quantum computing?

This being a graduate course means that if you are here then you probably already kind of know an answer to this question. For instance, you've probably heard of something called Shor's factoring algorithm, which is supposed to be pretty cool, but maybe you're not entirely sure why. Or a little less specifically, maybe you've heard that quantum mechanics can be exploited to speed-up certain computations. If either of these sounds like you and you want to learn the details, then this aims to be exactly the course for you. This is all to say that I won't spend too much time answering this question now, and will generally let the things we learn in the coming weeks speak for themselves.

That said, let me pay some lip service to the fact that this question—why quantum computing?—actually has *several* answers, which reflects the fact that quantum computing currently sits at a confluence of several disparate fields—physics, computer science, engineering, and yes, sometimes, even mathematics.

I will give brief answers to this question from the perspective of each these subjects, but to be entirely frank, my two cents is that from the question of pure science the most interesting answer is the one most closely related to computer science (and, in fact, Shor's algorithm). So let me start there.

1.1 Why computer scientists should care about quantum computing

1.1.1 Threatening the extended Church-Turing thesis

One of the first rigorous formalizations of the idea of a "fully programmable" computer was the universal Turing machine. Other formalizations are possible (e.g. using Church's λ-calculus) and in fact today most electronic devices we interact with in our day-to-day lives are generally understood to be "universal computing machines," meaning that, given access to enough working memory/RAM, they are able–in principle–to execute arbitrary computer code. Moreover, give any two reasonable notions of what it means for soemthing to be "computable," one can usually show that that they are equivalent. These observations led to the formalization of a (hypo)thesis or principle, called the Church-Turing thesis: anything that is "computable" is computable using a Turing machine. In other words, any two ways of defining "computable" are equivalent. [1]

In fact, given two models of computation, often something stronger can be shown: anything that can be *efficiently* computed in one model can be *efficiently* computed in the other model. This is called the **extended Church-Turing thesis**. Here by "efficient" I mean what computer scientists usually mean, namely: can be computed using an algorithm with running time upper bounded by a polynomial in the size of the input to the problem.

[1] Let me note that it's not really clear how to formulate the Church-Turing thesis as a rigorous mathematical theorem. Indeed, the main issue is how to define "computable" in the first place, and since "Turing machines" give one definition that seems sensible to me, I prefer to think of the "thesis" as really being a reality check that any *other* proposed definition of computable needs to satisfy in order to count as being realistic. For example, if you insist on using bouncing billiard balls to define "computable" (this is actually not crazy, see Figure 3.13 in Nielsen & Chuang), then I will insist that you show that this definition of computable is equivalent to the definition using Turing machines before I agree that it is sensible.

The main reason quantum computers are so fundamentally interesting is that they threaten the validity of the extended Church-Turing thesis. That is, there are certain things we believe a quantum computer can do more quickly than *any* non-quantum computer—even a supercomputer. In other words, the specific model of computation one employs—in particular, whether or not it is a classical[2] computer or a quantum computer—effects the definition of "efficiently computable."

Let me stress that I am not saying that *all* problems can be solved more quickly on quantum computers. (This is way more than necessary to disprove the extended Church-Turing thesis!) Rather, it is believed there are *some* problems that can be solved more quickly—that is, with a superpolynomial improvement in running time—on quantum computers than on classical computers. And this has nothing to do with the specifics of the engineering of the computers (e.g. clock speeds), beyond the assumption that one uses quantum mechanics and the other does not.

Shor's factoring algorithm is perhaps the most compelling evidence of this point. Recall the factoring problem: given an integer N expressed in binary (which only requires $O(\log N)$ many bits to express!), determine the prime factorization of N. There is no known classical algorithm that can solve this problem efficiently, *i.e.* in a running time bounded by $\text{poly}(O(\log N))$. However, as Peter Shor showed, there is a quantum algorithm that factors integers efficiently! Understanding this algorithm is one of our main goals this semester.

That said, there is good reason to be skeptical of Shor's algorithm. I don't mean that there is something about it that is wrong. I mean that there are certain assumptions we have to make in order to use Shor's algorithm as proof that the extended Church-Turing thesis is correct, which a skeptic might take issue with.

First, and most obviously, is that we have to assume that there does not exist a classically efficient algorithm for factoring numbers. As of this writing, it is certainly within the realm of possibility (without violating assumptions such as $\mathsf{P} \neq \mathsf{NP}$). Moreover, a skeptic might point to the example of *primality testing*. In this problem, rather than factor a given integer N, we simply have to decide—YES or NO—is N prime? For a few decades, the only known efficient algorithms—such as the first, the Miller-Rabin primality test—wre *probabilistic* algorithms. And then, somewhat out of nowhere in 2002, a deterministic primality test was discoverd by Agrawal, Kayal and Saxena. So, in analogy, a skeptic might say that it's just a matter of time until someone discovers a classically efficient algorithm (whether deterministic or probabilistic) for factoring.

To this, my immediate response would have to be, sure, that's possible.

However, de-randomization is very different from de-quantization! In fact, it is conjectured by many computer scientists that *all* efficient classical probabilistic algorithms can be "de-randomized," that is, replaced with efficient *deterministic* algorithms. (This is closely related to the existence of one-way functions and certain cryptographic assumptions.) And so the AKS derandomization is really just one example confirming a much more general expectation.

On the other hand, there has been an robust thread of quantum mechanical skepticism going back for 100 years—that is, for as long as quantum mechanics has been consciously conceived by humans! I won't have time to cover these things in any detail,[3] but the basic gist is that there have been many people over the years who have tried to "explain away" quantum mechanics either by recasting it in classical ideas or by showing that it can't possibly be correct. And basically all of these people failed. In some cases, (e.g. John Bell) these skeptics actually wound up formulating experimentally testable hypotheses that they had intended to be used to debunk quantum mechanics but instead, these experiments just confirmed quantum mechanics even more (and led to Nobel prizes!)!

So while it is admittedly within the realm of possibility that all efficient quantum algorithms can be "de-quantized" to yield efficient classical algorithms, this would go against the grain of a century of evidence.

So what's the second assumption we need to interpret Shor's algorithm (or quantum computing more generally) as a threat to the extended Church-Turing thesis? Well, it's pretty simple—it's the assumption that we actually have a quantum computer!

The skeptic's argument here is pretty simple: "show me!" That is, factor a number on a quantum computer, and show me the result.

This is actually more compelling than it sounds! Since it's discovery in 1994, no one has been able to implement Shor's algorithm on a quantum device that is capable of factoring any number bigger than 21.

The problem is that real world quantum computers are *noisy*, and we need to implement *error-correction* on them

[2]Here and throughout the course, "classical" is simply a synonym for "not quantum."

[3]If you want to know more, a good place to start is Aaronson's breezy undergraduate text "Quantum computing since Democritus." I also encourage you to take a quantum mechanics class from the physics department.

to use them. And no one has figured out how to do that yet, at least, not enough to factor even embarrassingly small numbers like 35 using Shor's algorithm.

We'll spend some time at the end of the semester talking about these issues and some of the theory that has been developed to overcome them (in theory!).

On a related note, let me conclude this discussion by noting that the (non-extended) Church-Turing thesis is not threatened by quantum computers. That is, anything that is computable using a quantum computer is also computable (perhaps more slowly!) on a classical computer. I should be able to say something rigorous about this in a couple of weeks.

1.1.2 Crypto

Shor's algorithm breaks the hardness assumptions of both factoring and discrete logarithm, and thus quantum computers threaten the security of many of the most common public key cryptosystems (RSA, Diffie-Hellman, elliptic curves,...). This motivates the study of *post-quantum cryptography*, the basic goal of which is to build cryptosystems that are secure even when they are attacked by an adversary with a quantum computer. There are two ways to do this.

First, you could chose to bring the big guns: *quantum* cryptography, which, despite its simplicity (we may discuss it later in the semester) has the benefit of being *provably 100% secure*. However, there are two drawbacks: it can be DDOS'd and it requires a quantum internet that connects the two parties hoping to establish a secure channel.

Alternatively, you could be much more practical: design a *classical* cryptosystem that is nevertheless quantum secure. Since there is nobody who expects all classical computing devices will be replaced with quantum devices anytime soon, this latter goal is quite important for ensuring all of our classical networking remains secure in the near term. The difficulty is identifying cryptographic primitives that are believed to be quantum secure. NIST is in the final stages of approving some of these. Google it if you want to learn more!

1.2 Why physicists should care about quantum computing

It is hard to overstate how influential "computational thinking" has become in physics.

This is especially the case in the study of quantum many body systems and condensed matter physics (e.g. systems consisting of lots and lots of interacting electrons) . For instance, there is currently developing a consensus that the definition of "(gapped) quantum phase of matter" (that is, certain quantum analogs of "solid," "liquid" and "gas") should have something to do with the computational complexity of converting the quantum states in one of the systems to states in the other system!

There have also been some recent fascinating connections drawn between high energy physics (read: quantum gravity) and quantum information. For instance, the conjectural Ryu-Takayanagi formula in the study of the holographic principle (that is, the AdS/CFT correspondence) relates the entanglement entropy in a "boundary" CFT (a quantum informational property) to the geometry in a "bulk" AdS space (a quantum gravity property). Other wild ideas include the "firewall paradox" and the principle that blackholes should (provably!) be able to scramble quantum information faster than any other physical system.

1.3 Why mathematicians should care about quantum computing

See the answer I gave for computer scientists! I think all mathematicians can agree that the fundamental question "what does it mean to compute"—which the Church-Turing thesis and quantum computation dig into—is worth exploring. Moreover, I would argue that there is a lot of beautiful "pure math" lurking in quantum computing.

Nevertheless, if this isn't the cut of your jib, then I'll give you a high profile example of the mathematically fundamental stuff lurking here: Ji, Natarajan, Vidick, Wright, and Yuen gave a "quantum computational" solution to Tsirelson's problem (a fundamental question in quantum information), which, by work of Junge and several others implies a negative solution to the Connes embedding problem (which has been a core problem in the study of von Neumann algebras—arguably "pure math" stuff—since the 1970s).

1.4 Why engineers should care about quantum computing

You may have noticed that quantum computers don't *really* exist yet, at least not in the generality that we would like (and will pontificate about in this class). There's a massive real-world engineering challenge lurking here, namely, to deploy quantum computers with "scalable" error-corrected qubit memories and universal, fault-tolerant operations. Achieving this will require significant advances in software, hardware, and everywhere in between, concerning essentially every component of the machines (e.g., how can we build better dilution refrigerators that enable us to carry extremely high-bandwidth signals in and out of them?). There are also near-term "practical" engineering questions, related to understanding how useful our current noisy quantum devices can nevertheless be.

A little less practically, if "computational thinking" has become essential in physics, then conversely we might say that "physical thinking" is essential in (electric) engineering. This is of course no surprise, but what I'm really getting at is that *quantum* physics is going to become more and more essential in certain domains.

Here's a fun little thought experiment. Recall Moore's law, which basically says that the density of transistors on an integrated circuit doubles every two years, at fixed cost. Of course, this is not a law of physics, it was simply an empirical observation that Gordon Moore made early in the developemt of microchip mass production. However, what if it were a physical law? Then within a few decades, we will be building transistors that are small enough to be "primarily" quantum mechanical objects!

2 Priming your brains: quantum mechanics as a "generalized" or "quantum" theory of probability

Before we can give a rigorous definition of "quantum computation," we all need to make sure we understand quantum mechanics! Since this is a CS class, I don't imagine most of you have taken a quantum mechanics class. Moreover, a rigorous approach to quantum mechanics requires comfort with some basic—but perhaps unusual—definitions and facts in linear algebra. So, we will spend this week beefing up on our linear algebra, before covering the axioms of quantum mechanics next week. Then in Week 3, we should be able to define "quantum computation."[4]

Having just said that, I want to prime your brains with a *little* piece of perspective that will hopefully allow you to tolerate the math this week.

As we will see next week, quantum mechanics—that is, the *axioms* of quantum mechanics—are essentially a set of rules saying that any quantum physical system must behave according to a certain set of rules that generalize the usual rules of probability. In fact, I prefer to think of the axioms of quantum mechanics as a set of axioms that specify a well-defined mathematical theory of "quantum probability." The main weirdness about the mathematics of "quantum probability" is that it looks quite different from usual probability theory, since it is defined using linear algebra over the set of complex numbers $\mathbb{C}$! For example, any complex number $z \in \mathbb{C}$ with $|z| \leq 1$ can appear as a "quantum probability," more precisely, as something called an "amplitude" of a (normalized) quantum state.

In particular, quantum computers are always probabilistic in nature. But much more is going on due to "quantum entanglement." I can't make that precise right now, but you'll see what I mean.

In fact, we can draw up a fairly precise analogy between the usual structures and laws of probability, and the structures and axioms of quantum mechanics. Check out Figure 1. Our goal now is to define all of the mathematics involved in the right column.

3 Linear algebra review

As I just said, our current goal is to review the mathematical definitions and theorems that are pertinent to formulating and understanding the axioms of quantum mechanics. We should get to the axioms on Tuesday of next week. Let's begin.

[4] Of course, I could just draw some examples right now, but considering how much ground I want to cover this semester, I am choosing to avoid the "spiraling in" approach to teaching this stuff and just drive straight to the generalities. This means that you will have to suffer through the math this week and the physics next week. That said, I have assigned all of Chapter 1 in Nielsen and Chuang as reading, and that will warm you up to *a lot* more than I have time for today.

Classical Probability	**Quantum mechanics**
finite sample space, i.e. finite set Ω	finite-dimensional Hilbert space $\mathscr{H}$
probability: $0 \leq p \leq 1$	amplitude: $z \in \mathbb{C}, \lvert z\rvert \leq 1$
distribution: $D : \Omega \to [0,1]$	non-zero vector $\lvert\psi\rangle \in \mathscr{H}$
random variable: $X : \Omega \to S$	observable: self-adjoint operator $A : \mathscr{H} \to \mathscr{H}$
outcome: $s \in S$	outcome of measuring A: an eigenvalue of A
symmetry of Ω: a bijection $\Omega \to \Omega$	symmetry of $\mathscr{H}$: a unitary transformation $U : \mathscr{H} \to \mathscr{H}$
sample space of a two part system: $\Omega = \Omega_1 \sqcup \Omega_2$	state space of a two part quantum system: $\mathscr{H}_1 \otimes \mathscr{H}_2$
law of total probability	Born rule

Figure 1: An analogy between classical (that is, non-quantum) probability and quantum mechanics.

3.1 Hilbert spaces

Definition. *Recall that a* vector space $(\mathscr{H}, +, \cdot)$ *over the complex numbers* $\mathbb{C}$ *is a set* $\mathscr{H}$ *together with two binary operations*

$$\begin{aligned} + : \mathscr{H} \times \mathscr{H} &\to \mathscr{H} \\ (x,y) &\mapsto x+y \end{aligned}$$

and

$$\begin{aligned} \cdot : \mathbb{C} \times \mathscr{H} &\to \mathscr{H} \\ (z,x) &\mapsto z \cdot x \end{aligned}$$

called vector addition *and* scalar multiplication*, respectively, that satisfy the following axioms:*

1. $+$ *is associative:* $(x+y)+z = x+(y+z)$ *for all* $x,y,z \in \mathscr{H}$.
2. $+$ *has an identity: there exists an element* $\mathbf{0} \in \mathscr{H}$ *such that* $\mathbf{0}+x = x = x+\mathbf{0}$ *for all* $x \in \mathscr{H}$.
3. $+$ *has inverses: for each* $x \in \mathscr{H}$ *there exists an element* $(-x) \in \mathscr{H}$ *such that* $x+(-x) = (-x)+x = \mathbf{0}$
4. $+$ *is commutative:* $x+y = y+x$ *for all* $x,y \in \mathscr{H}$.
5. $\cdot$ *is "associative:"* $z_1 \cdot (z_2 \cdot x) = (z_1 z_2) \cdot x$ *for all* $z_1, z_2 \in \mathbb{C}$ *and* $x \in \mathscr{H}$.
6. $1 \in \mathbb{C}$ *is an identity for* $\cdot$*: for all* $x \in \mathscr{H}$, $1 \cdot x = x$.
7. *scalar addition distributes over scalar multiplication: for all* $z_1, z_2 \in \mathbb{C}$ *and all* $x \in \mathscr{H}$, $(z_1+z_2) \cdot x = (z_1 \cdot x) + (z_2 \cdot x)$
8. *vector addition distributes over scalar multiplication: for all* $z \in \mathbb{C}$ *and all* $x_1, x_2 \in \mathscr{H}$, $z \cdot (x_1+x_2) = (zx_1) + (zx_2)$.

For what it's worth, the first three conditions say that $(\mathscr{H}, +)$ is a group, and the fourth says it's a *commutative* (or *abelian*) group. Condition 5 is a basic sanity check that says we don't need to worry about parentheses in scalar multiplication. The remaining conditions say that the scalar multiplication plays nice with the vector addition. As a little exercise, you might use the axioms to convince yourself that $(-x) = (-1) \cdot x$ for all $x \in \mathscr{H}$.

Note that we will usually suppress the $\cdot$ in scalar multiplication, and just write zx for $z \cdot x$.

Recall that the complex conjugate of the complex number $z = a+ib$ is the complex number $z^* = a-ib$. Given a complex number $z \in \mathbb{C}$, we will write $z \geq 0$ to mean that z is a positive real number. (Recall that $\mathbb{R} \subset \mathbb{C}$.) There is okay because there isn't really a meaningful total order on the set of all complex numbers, only the real numbers.

Definition. *A* complex inner product space $(\mathscr{H}, +, \cdot, \langle -,- \rangle)$ *is a vector space* $(\mathscr{H}, +, \cdot)$ *over the scalars* $\mathbb{C}$ *together with a function* $\langle -,- \rangle : \mathscr{H} \otimes \mathscr{H} \to \mathbb{C}$ *called the* complex inner product *such that*

1. *for all* $z_1, z_2 \in \mathbb{C}$, $x, y_1, y_2 \in \mathscr{H}$,
$$\langle x, z_1 y_1 + z_2 y_2 \rangle = z_1 \langle x, y_1 \rangle + z_2 \langle x, y_2 \rangle.$$

2. *for all* $x, y \in \mathscr{H}$,

$$\langle x, y \rangle = \langle y, x \rangle^*$$

where $*$ *denotes complex conjugation.*

3. *for all* $x \in \mathscr{H}$, $\langle x, x \rangle \geq 0$, *with* $\langle a, a \rangle = 0$ *if and only if* $a = \mathbf{0}$, *where* $\mathbf{0}$ *is the zero vector of* $\mathscr{H}$.

In particular, condition 3 implies that $\langle x, x \rangle$ is a *real* number.

Definition. *Let* $(\mathscr{H}, +, \cdot, \langle -, - \rangle)$ *be a complex inner product space. The* length *of a vector* $a \in \mathscr{H}$ *is* $||a|| = \sqrt{\langle a, a \rangle}$. *The* distance *between* $a, b \in \mathscr{H}$ *is* $||a - b||$.

Theorem 1 (Cauchy-Schwarz inequality)**.** *If* $\mathscr{H}$ *is a complex inner product space and* $a, b \in \mathscr{H}$, *then* $\langle a, b \rangle \langle b, a \rangle \leq ||a||^2 ||b||^2$.

You can find a proof in the book.

Corollary 2 (Triangle inequality)**.** *If* $\mathscr{H}$ *is a complex inner product space and* $a, b \in \mathscr{H}$, *then*

$$||a + b|| \leq ||a|| + ||b||$$

In particular, every complex inner product space is a metric space.

This is fairly easy to prove using Cauchy-Schwarz.

Definition. *A complex inner-product space is a* Hilbert space *if the metric induced by* $\langle -, - \rangle$ *is* complete.

Here "complete" means "Cauchy sequences converge." If you don't know what this means, do not worry! The most important thing to know is the following: *all* finite-dimensional complex inner product spaces are automatically complete. Thus, all finite-dimensional complex inner product spaces are Hilbert spaces!

We will often abuse notation and just write $\mathscr{H}$ to denote a Hilbert space space, and we will often just say "inner product" to mean "complex inner product." The first two conditions on the inner product are often summarized by saying that it is "sesquilinear." The third condition is called "positive definiteness."

3.2 Types of linear transformations on Hilbert spaces

There are two especially important types of linear transformations on Hilbert spaces: Hermitian operators, and unitaries. They are both defined with respect to a construction called the *adjoint*.

Definition. *Let* $A : \mathscr{H} \to \mathscr{H}$ *be a linear map. The* adjoint *of* A *is the (unique) linear transformation* $A^* : \mathscr{H} \to \mathscr{H}$ *that satisfies the following property:*

$$\langle A^* x, y \rangle = \langle x, Ay \rangle$$

for all $x, y \in \mathscr{H}$.

Lemma 3. A^* *is well-defined, and* $(AB)^* = B^* A^*$.

Proof. Exercise! □

Note that if we express A as a matrix with respect to an orthonormal basis of $\mathscr{H}$, then A^* is simply the conjugate-transpose of A. (Exercise!)

Definition. *A is*

- Hermitian *(self-adjoint) if* $A = A^*$.
- unitary *if* $A^* = A^{-1}$
- normal *if* $AA^* = A^* A$ *(that is, A commutes with its adjoint)*

Note that Hermitian or unitary operator is normal.

Lemma 4. *$U : \mathscr{H} \to \mathscr{H}$ is unitary iff $\langle Ux, Uy\rangle = \langle x, y\rangle$ for all $x, y \in \mathscr{H}$.*

Thus, unitary operators are exactly the transformations of $\mathscr{H}$ that preserve the inner products of all pairs of vectors.

Definition. *$A : \mathscr{H} \to \mathscr{H}$ is* unitarily diagonalizable *if there exists an orthonormal basis of $\mathscr{H}$ such that the matrix of A is diagonal in that basis.*

Of course, the basis will consist of eigenvectors of A and the diagonal entries will be the corresponding eigenvalues.

Theorem 5 (Spectral theorem in finite dimensions)**.** *Let $\mathscr{H}$ be a finite-dimensional Hilbert space. Then an operator $A : \mathscr{H} \to \mathscr{H}$ is normal if and only if A is unitarily diagonalizable.*

I'll discuss the proof next class. The next corollary is on your homework.

Corollary 6. *A is Hermitian iff it is unitarily diagonalizable with* real *eigenvalues. A is unitary iff it is unitarily diagonalizable with all eigenvalues of length* 1.