CS 59300 IQC (20557)/MA 59500 007 (20434)
Intro to Quantum Computing
Spring 2024
Syllabus

Course Description

An introduction to quantum computation focused primarily on foundations, theory, and rigor, rather than specific hardware implementations or heuristic applications. We will begin with the axioms of quantum mechanics and the most common formulation of quantum computation based on quantum circuits. We will then develop the core primitives in the quantum algorithms toolkit (such as quantum Fourier transforms, phase estimation, and Trotterization/quantum simulation) and establish some elementary complexity-theoretic results (including some oracle separations, and various lower and upper bounds), as well as work through the crown jewel of quantum algorithms to date—Shor's factoring algorithm. Along the way, we will see some of the more curious aspects of quantum information facilitated by quantum entanglement (such as Grover search, quantum teleportation, superdense coding, Bell violations). The last portion of the course will develop the basic theory of quantum error-correcting codes and the fault tolerance problem.

In particular, you may want to note that I do *not* plan to cover quantum optimization, quantum machine learning, or post-quantum cryptography in any depth (if at all).

Course Info

- 3 credit hours
- Modality: face-to-face (BoilerCast recordings and non-interactive livestream may be made available)
- Lecture schedule: Tuesdays and Thursdays from 10:30am to 11:45am
- Location: Physics Building 111
- Textbook: *Quantum Computation and Quantum Information*, Nielsen and Chuang.
- No exams. Homework every 1-2 weeks. Good faith participation required. All students must contribute to the shared course notes (see below).
- Prerequisites: some first-year and most second-year graduate students in CS, physics or mathematics should be comfortable in the class. Familiarity with at least one of the following at the level of a first year graduate student will be expected of all students: CS theory, quantum mechanics, abstract algebra, functional analysis, linear algebra. Please inquire with the instructor if you have any questions about whether this course is appropriate for you.

Learning Outcomes After completing the course, successful students should be able to do the following:

1. Formulate the axioms of quantum mechanics and all relevant definitions.
2. Explain what it means for a quantum computer to solve a problem, as well as what it means for a quantum computer to *efficiently* solve a problem.
3. Appreciate the essential role that measurement plays in quantum computation.
4. Describe several examples of impressive things that quantum computers can do when compared to classical computers.
5. Discern some of the limitations of quantum computers, both theoretical and practical.
6. Understand the fault tolerance problem, and how quantum error-correction works to partially address it.

Instructor Info

- Name: Eric Samperton
- Email: eric@purdue.edu
- Office: Mathematical Sciences Building 402
- Office phone: 765-494-1937
- Contact preferences: in-person > email > office phone > Brightspace. I do not expect to check messages on Brightspace regularly. I will strive to reply to all emails within 24 hours, although I will not usually reply to emails over the weekend.
- Office hours: Thursdays fifteen minutes after class, 12:00pm to 1:00pm.

Teaching Assitant Info

- Name: Vatsal Jha
- Email: jha36@purdue.edu
- Office: TBD
- Office hours: TBD

Recommended Resources

- Additional textbook: *Classical and quantum computation*, Kitaev, Shen, and Vyalyi.
- Entertaining and thorough background: *Quantum computing since Democritus*, Aaronson.
- Regular and reliable access to BoilerCast recordings on Purdue's MediaSpace.

Assignments

- Homework: it will be due approximately every 1-2 weeks, for a total of somewhere between 6 and 16 assignments. You should expected the assignments to be longer and more frequent at the beginning of the course as we work through essential foundational material.
- Course notes contribution: students will rotate preparing typed notes (using LaTeX) for my lectures, to be posted on the course website for all to share. This must be done in a timely manner, so that your classmates may use the notes for reference. (How many times each student must do this will depend on course enrollment. If enrollment is high, then I may allow the preparation of a pedagogically-sound homework solution set (to be shared with the entire class) as a course note-taking contribution.)
- Participation: if you can't regularly attend class and engage with me by asking questions and contributing to discussions, then you should not enroll.

Course Schedule

Here is rough outline of the schedule of topics we will cover (subject to change):

- Weeks 1-2: algebra prelude. Hilbert spaces, unitary matrices, Hermitian matrices, groups (especially unitary groups, abelian groups). Axioms of quantum mechanics. What's really in a quantum state?
- Week 3: qubits, qudits, and quantum circuits. Reversible computing concerns (reversible dilation of classical Boolean functions, controlled gates, uncomputation). Solovay-Kitaev theorem. BQP and its relation to other complexity classes (e.g. Bernstein-Vazirani).

- Weeks 4-5: cute quantum stuff. Grover search, quantum teleportation, superdense coding, Bell violations. (This last one is pretty serious actually! It won a Nobel in 2022!)
- Week 6: Various no-go results. No cloning. Grover lower bounds. Holevo bound.
- Week 7: algebra interlude. Quantum Fourier transform and quantum phase estimation.
- Weeks 8-9: Hamiltonian complexity basics: quantum simulation, Trotterization, QMA and the local Hamiltonian problem.
- Weeks 10-11: Shor's algorithm. The hidden subgroup problem. Comments on graph isomorphism etc.
- Week 12: foundations of quantum error correction.
- Week 13: Pauli stabilizer codes. CSS codes. Gottesman-Knill theorem.
- Week 14: toric code (aka surface code) and its cousins.
- Time-permitting: a brief peak at codes that support BQP-universal fault-tolerant operations from topological quantum field theories (e.g. the doubled Fibonacci theory TQFT).

Maybe more importantly, I want to be forthright that I expect the pacing of the course to be a little uneven. Some weeks will be heavy, and some will be light. **Here is a rough outline of the vibe I expect**:[1]

- Week 1: ugh lots of definitions. What's a tensor product again? Eigenwhat now? Why are there so many exercises on the homework?
- Week 2: As John von Neumann once said: "You don't understand quantum mechanics, you just get used to it." Why are there so many exercises on the homework?
- Week 3: wow, quantum circuits are actually pretty simple to define! And there are some (admittedly rather contrived) problems that make it easy to see that they're doing something really weird that is hard to emulate on a classical computer!
- Weeks 4-5: some easy payoffs and insights, but (except for Bell violations), most of these things are kinda gimmicky
- Week 6: Sure, sometimes getting our noses rubbed in bad news is important, but can we just get to factoring yet? At least the homework is getting shorter now.
- Week 7: "can't we just *define* the Fourier transform to be this matrix?" "I suppose but I staunchly refuse!"
- Weeks 8-9: oh yeah, I guess quantum computers *should* be helpful for physics? What's a Hamiltonian again?
- Weeks 10-11: we're really rippin' now, dudes!
- Week 12: record scratch, back to a morass of definitions.
- Week 13: symplectic what?
- Week 14: ooo pretty pictures
- time-permitting: lots and lots of pretty pictures (is there not any homework this week?)

[1] Weekly vibe checks will be undertaken in order to calibrate this class for future semesters.

Grading Scale and Policies
Depending on the size of the class, I may have some flexibility with late homework, but don't count on it unless you ask me in advance, are sick, or have some kind of emergency.

Your final course raw score (as a percentage) will be computed using the following weights:

Homework	60%
Course notes contribution	20%
In-class participation	20%

Your final course letter grade will be determined by converting your raw score into a letter grade, with lower cutoffs for each letter at least as generous as the following:

A+	97%
A	93%
A-	90%
B+	87%
B	83%
B-	80%
C+	77%
C	73%
C-	70%
D+	67%
D	60%

At the end of the semester these lower cut-offs may be adjusted upward. (That is, the final course grades may be "curved.")

Attendance Policy
Please do not come to class if you are sick with a messy cold, COVID, the flu, or some other temporary and highly-transmissible illness. Please note that I am also happy to be flexible about your attendance if you have any unexpected emergencies or important travel (e.g. to present at a conference). In these events, you should try to let me know in advance whenever possible, and you should catch up on class by watching BoilerCast recordings. Having said all this, as noted above, if you can not regularly attend class and engage with me by asking questions and contributing to discussions, then you should not enroll.

Academic Integrity
You are welcome to use any textbooks, research papers, or other notes to solve homework problems, but you must use your own words when wrtiing up your solutions, and you must cite any sources you used; these policies also apply to the use of generative AI such as ChatGPT. On the other hand, "homework help services" (that is, professional cheating companies) such as Chegg, CourseHero, etc, are not allowed at all; any use I discover will result in an immediate F for the course.

Students with Disabilities
Please submit DRC requests for accommodations as soon as possible.

Other Policies
This course is subject to all University Policies and Statements, the most pertinent of which can be found on our Brightspace.

In extraordinary circumstances, this syllabus may be amended or changed as necessary in order to facilitate fair learning and grading.