Lecture 1: Quantum simulation
Reading: Quantum simulation, I. M. Georgescu, S. Ashhab, and Franco Nori, Rev. Mod. Phys. 86, 153 (2014).

1 Introduction

Recent advances in physical hardware for various types of quantum devices (e.g. quantum annealing as in DWave or gate-based quantum computers by Google, IBM, Rigetti, etc) have received tremendous attention in the past 5 years or so [1]. Of course, the quantum information and computing fields have been active for around 40 years since the original papers by Benioff [2], Feynman,[3] and Deutsch [4] describing computation with quantum rather than classical degrees of freedom. The past 20 or so years have seen substantial increases in the size of the quantum information community following the introduction of Shor's algorithm for factoring prime numbers with exponential speedup compared to the best known classical methods.

Partly because of that seminal result, the QI community has attracted scientists from many fields including mathematics, computer science, and physics, to work on problems of what I would argue is a computer science flavor, e.g. Simon's algorithm for identifying hidden subgroups, combinatorial optimization, cryptography, communication and others. Caltech has a number of courses on these topics (e.g. quantum cryptography, and a general quantum computing course). Quantum information is also gaining attention for the applications of its concepts to black holes.

What is not as well covered in terms of courses is the use of QC for one the earliest suggested purposes: as a tool to study and learn about other quantum systems, or a quantum simulator. This concept was originally proposed by Benioff and Feynman as a way to overcome the exponential explosion in the size of Hilbert space with the number of particles, which is the origin of the intractability of exactly describing quantum systems with classical computers. Further, much of the work performed by the QI community assumes the existence of an error-corrected quantum computer with non-trivial numbers of logical qubits. Presently, of course, we have relatively small numbers of imperfect physical qubits operated on by imperfect gates. What problems in physics might we solve with these early-stage quantum computers?

This course aims to provide the relevant background for quantum computing and then present a selection of recent results of algorithms that may be suitable for near-term quantum computers. Whether they will actually prove useful to solve a physically relevant problem that could not be solved by other means is not yet clear. But, we should have answers in the 5 year timeframe!

1.1 Problem statement

Our general goal is to solve the governing equations for a quantum system and compute relevant quantities from this solution such as identifying the lowest energy state (the ground state), thermal averages, correlation functions, and so on. As a common example, a problem statement is to solve the Schrodinger equation:

$$i\frac{d\left|\psi\right\rangle}{dt} = H\left|\psi\right\rangle \tag{1.1}$$

where H is the Hamiltonian operator and E is the energy of the eigenstate (we set $\hbar = 1$). The general solution of this equation is:

$$\left|\psi(t)\right\rangle = e^{-iHt}\left|\psi(0)\right\rangle = U(t,0)\left|\psi(0)\right\rangle \tag{1.2}$$

If we have N spin-1/2 particles, we need 2^N complex numbers just to specify the state. To figure out the time evolution, we need to exponentiate a $2^N \times 2^N$ matrix. The threshold for the exact calculation of this time evolution on a classical computer is around 40 spins, which in single precision requires 4 TB of memory. For 80 spins, we would need 3.8×10^{25} bits which is more information than has ever been stored by humans! [5]

Now, a standard approach to overcome computational challenges associated with large phase spaces is using stochastic methods like quantum Monte Carlo. Although this method does work well for some problems, mapping fermions and some other systems to the classical stochastic problem leads to the presence of nonpositive semi-definite weights, known as a sign problem. Solving this problem is NP-hard, meaning a general solution is classically intractable. [6]

However, in many cases there are clever approximate methods that make use of the physical structure of the system. For instance, for weakly correlated systems we know that an approximate wavefunction for the system (e.g. a Slater determinant) or density-based approaches like density functional theory do very well. As another example, we know that for 1D systems certain types of Hamiltonians have restricted entanglement, enabling their near exact, efficient classical description using the matrix product state ansatz. At this point in time, after nearly 100 years of study of the electron gas and other quantum systems, the remaining problems like the origin of high-Tc superconductivity, problems in particle physics, and so on, are hard problems that are not amenable to the above approximations. Progress in classical algorithms is certainly still advancing, particularly for relatively newer approaches like tensor networks, but it is natural to wonder about the application of another tool to these tough problems, e.g. a quantum simulator.

1.2 Quantum simulation

Quantum simulation is defined as the use of a controllable quantum system to emulate the interactions and evolution of another quantum system. Say the state of a system to be simulated is $|\psi\rangle$ and it evolves from $|\psi(0)\rangle$ to $|\psi(t)\rangle$ by unitary evolution. The propagator that implements this time evolution is

$$U = e^{-iH_{sys}t} \tag{1.3}$$

Now suppose we have a quantum simulator that is controllable: we can prepare $|\phi(0)\rangle$, implement a propagator U' with H_{sim}, and find a final state $|\psi(t)\rangle$ that can be measured with some observables. The original system can be simulated if a mapping exists between $|\psi\rangle$ and $|\phi\rangle$. Note that the key property of quantum simulators is their ability to store exponential information in polynomial physical memory. This property is certainly desirable not not sufficient for quantum simulation: we also need efficient methods to prepare initial states, evolve the system, and measure desired quantities with polynomial resources.

Analog quantum simulation (AQS)

A general purpose quantum computer is practically very difficult to build owing to decoherence of physical qubits, imperfect gates, and numerous other challenges. An alternate approach that has seen great success over the past 20 years is the use of quantum systems with some degree of control over the interactions of the quantum degrees of freedom to imitate other systems - an analog quantum simulator (AQS). The most obvious example of such a quantum simulator is an ultracold atoms confined in an optical lattice or by other means so that the atoms experience an external potential with some interactions. Fortuitously, the Hamiltonian describing the atoms in this system overlaps nicely with the Hubbard model, and the appropriate statistics (Bose or Fermi) can be selected by choosing the right type of atoms.

More precisely, in AQS H_{sys} is mapped directly onto H_{sim} which is partially controllable. As an example, consider that we have atoms in an optical lattice with interactions that can be tuned with external light pulses. A gas of interacting bosonic atoms in a periodic potential has the Hamiltonian:

$$H_{sim} = -J \sum_{\langle i,j \rangle} a_i^\dagger a_j + \sum_i \epsilon_i n_i + \frac{1}{2} U \sum_i n_i (n_i - 1) \tag{1.4}$$

which is very similar to a Bose-Hubbard model which is of interest in condensed matter physics. Therefore, phenomena related to e.g. superconductivity can be studied with ultracold atoms in an optical lattice as an AQS. Ultracold gases have been used in numerous papers to study a wide-range of phenomena, including quantum phase transitions, quantum dynamics, many-body localization, and other topics.

As another example, consider the Dirac equation in $1+1$ dimensions for a spin-1/2 particle with mass m:

$$i \frac{\partial \left| \phi \right\rangle}{\partial t} = (c\hat{p}\sigma_x + mc^2 \sigma_z) \left| \phi \right\rangle \tag{1.5}$$

The Hamiltonian of a single trapped ion in a bichromatic light field is:

$$H_I = 2\eta \Delta \bar{\Omega} \sigma_x \hat{P} + \Omega \sigma_z \tag{1.6}$$

where η is the Lambe-Dicke parameter, Δ is the spatial extent of the ground-state wavefunction, and $\bar{\Omega}$ is a parameter controlled by the light field. Noticing the similarity, we see that by tuning the light field we can study relativistic physics with a non-relativistic particle!

Clearly, these forms of analog quantum simulators are already quite capable, and their abilities continue to grow. Why would we like a general purpose quantum computer based on gates and qubits? We can identify a few reasons that are analogous to why our current digital classical computers are so powerful and displaced analog classical computers. First, analog quantum simulators are restricted in what Hamiltonians they can simulate and the experimentally accessible parameters of those Hamiltonians. For instance, atoms in an optical lattice naturally possess a Hamiltonian similar to that of the Hubbard model and so we are able to study the physics of that model. If we want to study an entirely different model, or have interactions that are not naturally mappable to the interactions of the atomic system, we need to do a lot more work to modify the setup of our simulator, if it is possible at all.

Second, general-purpose quantum computers offer a degree of control over the quantum state of the device and the interrogation of that state that is not possible with typical optical lattice setups. In an AQS, we can apply Hamiltonians to the atomic system, change parameters and induce dynamical processes, and observe those processes with imaging. In this typical procedure, we are unable to specify the quantum state of an individual atom nor measure arbitrary expectation values of operators for atoms in the lattice. Quantum computers thus offer an unparalleled degree of quantum control that in turn allows us to simulate quantum systems and measure observables that are not possible with AQS.

Digital quantum simulation

Digital quantum simulation is based on the circuit model we will discuss extensively in the course. The wavefunction is encoded in bit strings in a register of qubits. To perform time evolution, we apply U in a series of 1 and 2-qubit gates. Now, any unitary operation (e.g. U) can be expressed with a universal gate set of 1 and 2-qubit gates but not necessarily efficiently - an exact decomposition requires exponential circuit depth! However, finite local Hamiltonians can be efficiently simulated as shown by Lloyd (and we will discuss in the next lecture).

Finally, we need to measure the properties of the wavefunction at the end of the time evolution. Naively, we would say we want to fully characterize the wavefunction so that we can computer any property we want. Unfortunately this process, known as quantum state tomography, requires exponential resources. Therefore, we will not have access to the full wavefunction generally and will have to choose observables that can be efficiently measured with polynomial resources.

What problems can be treated on these computers? The standard Schrodinger equation is by far the most common and so we focus on it in this class. However, many other problems are being considered to be treated on a DQS (I will just call this a quantum computer, or QC), including the Sachdev-Ye-Kitaev model for quantum gravity, lattice gauge theories, cosmology, information scrambling in black holes, and more! The Hamiltonians of interest can take many forms, e.g. spin models like the Heisenberg and Ising models, approximate electronic structure models like the Hubbard model, full electronic Hamiltonians as occur in quantum chemistry and computational materials science, and others. As we will discuss in detail later, in general the computational complexity to find certain properties to some specified precision for these Hamiltonians is "hard" - e.g. NP complete classically and QMA-complete for quantum computers. So quantum computers will not be a magic bullet to solve interesting quantum problems - we will usually need some type of physical knowledge or approximation to make the problem tractable even on a quantum computer.

Interestingly, simulating the transient behavior of a quantum system with a given Hamiltonian is in the complexity class BQP (analogous to P for classical computers) meaning that it is tractable on a quantum computer. The same task is generally not possible on a classical computer due to exponential memory requirements. Therefore, any exact quantum computation on a classical computer is generically "hard" but certain calculations on a quantum computer are tractable. In addition, on a quantum computer we are able to store an exact quantum state with polynomially many qubits, and so our memory capabilities are dramatically enhanced compared to those of classical computers. These observations provide a motivation for trying to understand precisely how a quantum computer could be helpful to study a quantum system compared to what is possible with a classical computer.

1.3 State of present hardware for DQS

In the past 5 years, rapid development in the hardware for a general-purpose QC has occurred. Many companies are investing substantially in various architectures including superconducting qubits, ion traps, and others. Presently, these systems have on the order of tens of qubits. I will focus on the specifications for the superconducting qubit systems as I am most familiar with those. The typical coherence times of the qubits are on the order of 50 μs. Gates can be applied at a rate of 100 MHz and with fidelities of $> 95\%$ for 1 qubit gates and $> 90\%$ for 2 qubit gates. Readout errors are order of a few percent. Note that some companies report much better specs but their hardware is not publically available yet, and so the numbers are hard to confirm. In addition to the errors and decoherence challenges, cross-talk can be a significant source of inaccuracy as well. Cross-talk occurs if applying a gate to one qubit induces a change in the state of another qubit.

Considering all these numbers, it is clear that near-term devices can likely execute only order 50 gates before one is simply processing noise. The number 50 comes from either the decoherence time of the qubits and the rate at which gates can be applied, or the success probability of applying gates with a non-unity fidelity (e.g. $0.99^{50} = 0.6$, so after 50 gates we are close to getting the correct outcome only a bit more than half the time). These numbers are not likely to get orders of magnitude better, either. A key target for error correction is 99.9% fidelity for the two-qubit gate and better than that for 1 qubit gates and readout, with sufficiently small cross-talk. Even if those numbers are achieved, we can apply < 500 gates before noise dominates, a number that is certainly better than 50 but nowhere near the estimates of the number of gates needed for, e.g. quantum chemistry, which are on the order of millions. The real advantage of realizing those fidelities is that error correction with a surface code can be performed with around 1000 physical qubits per logical qubit.

These observations also imply that simply increasing the number of qubits is not helpful beyond a certain number, because qubits provide a quantum advantage only if they can be entangled. We can only generate enough entanglement for the overall system if we can apply a sufficient number of two-qubit entangling gates, which as we have just seen is limited in present devices. Further, actual devices typically have only nearest-neighbor connectivity, limiting which qubits can be entangled without additional SWAP gates being required to place qubits adjacent to each other. This consideration leads to the idea of "quantum volume," which describes the capability of a quantum device in terms of the number of qubits along with the number of gates that can be applied with sufficient fidelity, connectivity, and the number of operations that can be run in parallel. All of these factors determine what algorithms can be run on a given device.

Therefore, the challenge for doing physics on near-term quantum computers is finding how to obtain a result that cannot be simulated classically given all of these restrictions that are quite severe. It is entirely possible the answer is that no such applications exist with present devices. However, the hardware is improving rapidly, and it is not unreasonable to think that in 5 years that answer may be different. With that in mind, the goal of this course is to introduce recently introduced algorithms that solve quantum problems in chemistry and physics that could be run on near-term devices, but perhaps will not surpass classical capabilities right now. But we will be prepared when the hardware advances sufficiently that the situation changes.

References

[1] M. Mohseni, P. Read, H. Neven, S. Boixo, V. Denchev, R. Babbush, A. Fowler, V. Smelyanskiy, and J. Martinis, "Commercialize quantum technologies in five years.", Nature **543**, 171–174 (2017).

[2] P. Benioff, "The computer as a physical system: a microscopic quantum mechanical hamiltonian model of computers as represented by turing machines", J Stat Phys **22**, 563–591 (1980).

[3] R. P. Feynman, "Simulating physics with computers", Int J Theor Phys **21**, 467–488 (1982).

[4] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer", Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **400**, 97–117 (1985).

[5] I. M. Georgescu, S. Ashhab, and F. Nori, "Quantum simulation", Rev. Mod. Phys. **86**, 153–185 (2014).

[6] M. Troyer and U.-J. Wiese, "Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations.", Phys Rev Lett **94**, 170201 (2005).