CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Tuesday, February 27 - Lecture 8.1

Today's scribe: Nico

Agenda:

1. Quantum Simulation
2. The Heisenberg Model

Intended to cover the Ising model, but did not get to it.

1 Quantum Simulation

Given a "reasonable" description of a (discrete) quantum mechanical system, we expect that we can simulate the system on a quantum computer. This expectation comes from the classical case. We can simulate classical mechanical systems on classical computers pretty well.

In the classical setting, we have two main ways to describe dynamical systems:

Newtonian Mechanics: We have a description of a force vector $\vec{F}$ on space. To simulate time evolution/dynamics, we use Newton's equation

$$\frac{d}{dt}\left(m\frac{d\vec{x}}{dt}\right) = \vec{F}$$

which amounts to solving some differential equation.

Hamiltonian Mechanics: Instead of a force vector field, we are given a Hamiltonian $\mathscr{H}$ which is a function on phase space to $\mathbb{R}$. This yields Hamilton's equations:

$$\begin{cases} \frac{d\vec{q}}{dt} = \frac{\partial \mathscr{H}}{\partial \vec{p}} \\ \frac{d\vec{p}}{dt} = -\frac{\partial \mathscr{H}}{\partial \vec{q}} \end{cases}$$

In quantum mechanics, we use the quantum version of Hamiltonian mechanics. According to Schrödinger, for a quantum mechanical system with state space V, the time evolution is controlled by a (quantum) Hamiltonian $\mathscr{H}$, which is a Hermitian operator $V \to V$, thought of as the *energy observable*. The axioms of quantum mechanics tell us that the Hamiltonian generates the dynamics of a quantum mechanical system via Schrödinger's equation:

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \mathscr{H}|\psi\rangle$$

Thus, given an explicit description of $\mathscr{H}$, simulating the corresponding quantum mechanical system requires that we (exactly/numerically/approximately according to our goals) solve this differential equation.

In all of these cases, ideally we would like to be able solve these differential equations explicitly by finding exact analytic solutions. But this is generally not possible even classically (e.g. the three-body problem). So, instead of solving, we try to *simulate* the dynamics using a computer.

The simulation takes in some inputs (initial state, initial time, and accuracy $\delta > 0$) and yields an output which is an approximation of the time evolved state after some time t to within δ-accuracy. Classically, this can be done by **discretizing** using various methods (finite element method, etc.), but quantum mechanically, this is trickier. In fact, we don't really have any better of a way to store quantum states on a classical computer besides just storing

each amplitude in the computational basis individually. This makes simulation via discretizing states on a classical computer **extremely costly**.

Not only is discretizing states costly, but in general discretizing the Hamiltonian $\mathscr{H}$ is costly as well! Luckily, however, most Hamiltonians encountered in physics are *"local"*, i.e. they are made up of smaller Hamiltonians which only act on a bounded number of particles at a time.

Any Hamiltonian $\mathscr{H}$ may be decomposed in the following way:

$$\mathscr{H} = \sum_{S \subset \{x_1,\ldots,x_n\}} \mathscr{H}_S$$

where x_i is a qubit, and $\mathscr{H}_S$ is a Hermitian operator on S, extended to all of $\{x_1,\ldots,x_n\}$ by tensoring with the identity.

Example: Let $n = 6$ and let $S = \{x_1, x_2, x_4\}$, then we could let $\mathscr{H}_S$ be given by:

$$\mathscr{H}_S = X \otimes (X+Z) \otimes Id \otimes Z \otimes Id^{\otimes 2}$$

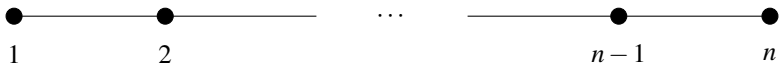
or

$$\mathscr{H}_S = CNOT \otimes Id \otimes (X-Z) \otimes Id^{\otimes 2}$$

To remedy these issues, Yuri Manin and Richard Feynman proposed that we simulate quantum systems on quantum computers.

1.1 The Heisenberg Model

Consider a "spin chain" comprised of tiny spins or magnets:



Each spin can be either up or down or in a superposition of both. We can write

$$|\text{up}\rangle = |1\rangle \qquad |\text{down}\rangle = |0\rangle$$

and thus we can say that our spin chain is really comprised of n qubits, and thus the state space of our spin chain is $(\mathbb{C}^2)^{\otimes n}$.

We imagine the neighboring spins are coupled in a "translation invariant way." We do this by choosing coupling constants $(J_x, J_y, J_z) \in \mathbb{R}^3$ and defining

$$\mathscr{H}_{\text{int}} = -\sum_{j=1}^{n-1} J_x X_j X_{j+1} + J_y Y_j Y_{j+1} + J_z Z_j Z_{j+1}$$

where $U_j U_{j+1}$ means $Id^{\otimes j-1} \otimes U \otimes U \otimes Id^{\otimes n-j-1}$.

We also may have an external magnetic field affecting our chain, say in the z-direction, described by

$$\mathscr{H}_{\text{ext}} = -\sum_{j=1}^{n} hZ_j, \qquad h \in \mathbb{R}$$

Then we may define

$$\mathscr{H} = \mathscr{H}_{\text{int}} + \mathscr{H}_{\text{ext}}$$

This is the *Heisenberg model* (for an (anti)ferromagnetic spin chain). As usual, the eigenvalues of $\mathscr{H}$ are the possible energy levels for our quantum "magnet" (which may not be a magnet at all!).

Some natural questions may arise:

- What are the energy levels for a given tuple of coupling constants?
- What is the lowest energy level (aka the ground state)?
- What about phase transitions?

We can get an intuition for the energy states by looking at a specific example:

Suppose $J = J_x = J_y = J_z$, then we call the resulting spin chain the **XXX chain**. When $J > 0$, then low energy configurations are "ferromagnetic" meaning the spins are aligned. I.e. our states might look like $\alpha|0\cdots0\rangle + \beta|1\cdots1\rangle$. When $J < 0$, the configurations are "anti-ferromagnetic" which means that the neighboring spins are opposite on average. You should be thinking of states that look like $\alpha|10\cdots10\rangle + \beta|01\cdots01\rangle$ (if n is even). For a more thorough analysis of this stuff, you might look up the phrase "Bethe ansatz."

1.2 Back to Simulation

Assume that we have a Hamiltonian on $(\mathbb{C}^2)^{\otimes n}$

$$\mathscr{H} = \sum_{k=1}^{L} \mathscr{H}_k$$

where L is a polynomial in n and each $\mathscr{H}_k$ is *c-local* meaning it is supported on at most c of the n qubits. $\mathscr{H}$ is called a *c-local Hamiltonian on n qubits*.

We would like to simulate the dynamics given by $\mathscr{H}$. I.e. given a state $|\psi\rangle$, a time t, and an accuracy $\delta > 0$, we want to use a quantum computer to prepare a state $|\phi\rangle$ which "approximates the time evolution," i.e. is close to the true time evolution $e^{it\mathscr{H}}|\psi\rangle$ in that

$$\left|\langle\phi|e^{it\mathscr{H}}|\psi\rangle\right|^2 \geq 1 - \delta.$$

Moreover, we'd like to do all of this as efficiently as possible in terms of the parameter n, t, and δ.

The main trick to accomplishing this is sometimes called "Trotterizing."

Lemma 1 (Trotter Formula)**.** *For A, B Hermitian, $t \in \mathbb{R}$,*

$$\lim_{n\to\infty} (\exp(iAt/n)\exp(iBt/n))^n = \exp(i(A+B)t)$$

Proof.

$$\begin{aligned}
\lim_{n\to\infty} (\exp(iAt/n)\exp(iBt/n))^n &= \lim_{n\to\infty} \left[\left(I + \frac{iAt}{n} + O(1/n^2)\right)\left(I + \frac{iBt}{n} + O(1/n^2)\right)\right]^n \\
&= \lim_{n\to\infty} \left[I + \frac{i(A+B)t}{n} + O(1/n^2)\right]^n \\
&= \lim_{n\to\infty} \left[\sum_{k=0}^{n} \binom{n}{k} \frac{1}{n^k} \big(i(A+B)t\big)^k + O(1/n)\right]
\end{aligned}$$

The last equality comes from using the Binomial theorem.

Note that

$$\binom{n}{k} = \frac{1+O(1/n)}{k!}$$

so

$$\begin{aligned}\lim_{n\to\infty} (\exp(iAt/n)\exp(iBt/n))^n &= \lim_{n\to\infty}\left[\sum_{k=0}^{n} \frac{1+O(1/n)}{k!}\big(i(A+B)t\big)^k + O(1/n)\right] \\ &= \sum_{k\geq 0}\frac{1}{k!}\big(i(A+B)t\big)^k \\ &= \exp(i(A+B)t)\end{aligned}$$

□

Claim 1. *Fix $c \in \mathbb{N}$. Given a c-local Hamiltonian on n qubits*

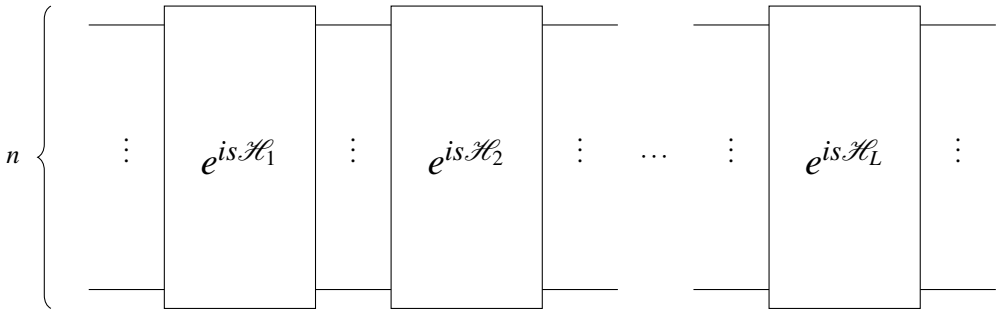
$$\mathscr{H} = \sum_{k=1}^{L} \mathscr{H}_k$$

, a state $|\psi\rangle$ and an accuracy $\delta > 0$, there is a quantum algorithm to prepare a state $|\phi\rangle$ such that

$$\left|\langle\phi|e^{it\mathscr{H}}|\psi\rangle\right|^2 \geq 1-\delta.$$

Moreover, the algorithm runs in time $O(\mathrm{poly}(1/\delta), t, n)$.

Proof Sketch. Since c is fixed, we might as well assume that our gate set includes *all* c-ary gates. Thus, we can assume that we can implement arbitrary c-ary gates in one unit of time. In particular, for all s, it takes only one unit of time to implement $\exp(is\mathscr{H}_k)$. Pick s appropriately (given t and δ) and let U be the following circuit, which we describe only schematically:



This is "schematic" in the sense that each "gate" here is really a subcircuit that should consist of a single c-ary gate on an appropriate choice of c of the n qubits, according to $\mathscr{H}_i$.

It is straightforward to verify that

$$\left\|\exp(it\mathscr{H}) - U^{\lceil t/s\rceil}\right\| < \delta.$$

This implies what we want. □