Ph/CS 219A
Quantum Computation

Lecture 17. Quantum Simulation

As Feynman foresaw nearly 40 years ago, the "natural" application for quantum computers is solving quantum problems.

We'll discuss two examples in this lecture. In both cases, we consider a quantum system with a local Hamiltonian: the Hamiltonian is a sum of terms, where each term acts nontrivially on a constant number of qubits, independent of the total system size. Local Hamiltonians are of particular interest, because we believe that the quantum systems encountered in Nature are described by such Hamiltonians. For such systems:

(1) Time evolution of a system with volume V can be simulated for time T with an error 1/poly(V), using a quantum circuit whose size is poly(V,T).

(2) Energy eigenvalues can be estimated to accuracy 1/ poly(V) using a quantum circuit with size poly(V), assuming we can efficiently prepare a quantum state whose overlap with the corresponding energy eigenstate is 1/ poly(V).

Updated Chapter 6 Lecture Notes have been posted on the course website. This lecture aligns with Sec. 6.8 of Chapter 6. Problems Set 4 is due on December 4.

Evolving with a local Hamiltonian

$H = \sum_a H_a, \quad H_a$ acts on k qubits where $k = \text{constant}, \quad \|H_a\|_\infty \le h = \text{constant}.$

Time-dependent Schroedinger: $|\psi(t)\rangle = U(t)|\psi(0)\rangle, \quad \frac{d}{dt}U(t) = -iH(t)U(t), \quad U(0) = I.$

Goal: $\| |\tilde{\psi}(t)\rangle - |\psi(t)\rangle \| < \delta = \text{constant}.$

Classical task: sample from probability distribution of outcomes for measurement of an efficiently computable observable, in the ideal final state.

Number of terms in Hamiltonian: $\binom{n}{k}$. If geometrically local: $O(n)$.

Evolve for time t as a sequence of t/Δ time steps each of width Δ.

Terms H_a do not commute. Approximate: $e^{-iH\Delta} \approx \prod_a e^{-iH_a\Delta}$. What's the error?

Error estimate

$$\exp(\sum_a A_a) - \prod_a \exp(A_a) = \left(1 + \sum_a A_a + \frac{1}{2}\sum_{a,b} A_a A_b + ..\right) - \prod_a \left(1 + A_a + \frac{1}{2}A_a^2 + ..\right)$$

$$= \left(1 + \sum_a A_a + \frac{1}{2}\sum_{a,b} A_a A_b + ..\right) - \left(1 + \sum_a A_a + \sum_a \frac{1}{2}A_a^2 + \sum_{a<b} A_a A_b + ..\right)$$

$$= \frac{1}{2}\left(\sum_{a<b} A_a A_b + \sum_{a<b} A_b A_a\right) - \sum_{a<b} A_a A_b + .. = \quad -\frac{1}{2}\sum_{a<b}[A_a, A_b] + ..$$

$$\Rightarrow \quad e^{-iH\Delta} - \prod_a e^{-iH_a\Delta} = \frac{1}{2}\Delta^2 \sum_{a<b}[H_a, H_b] + \ldots \quad \Rightarrow \quad \left\| e^{-iH\Delta} - \prod_a e^{-iH_a\Delta} \right\| = O(M\Delta^2 h^2).$$

(*M* terms in Hamiltonian, geometrically local case)

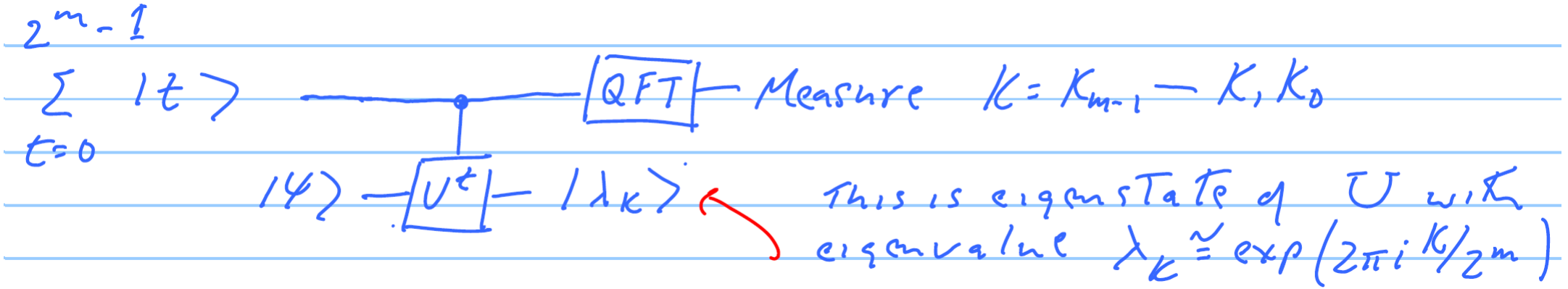
$$\Rightarrow \quad \delta = \frac{t}{\Delta} \times O(M\Delta^2 h^2) \quad \Rightarrow \quad \Delta = O\left(\frac{\delta}{h^2 Mt}\right) \quad \Rightarrow \quad L = O\left(\frac{Mt}{\Delta}\right) = O\left(\frac{h^2 (Mt)^2}{\delta}\right) = O\left(\frac{h^2 (nt)^2}{\delta}\right).$$

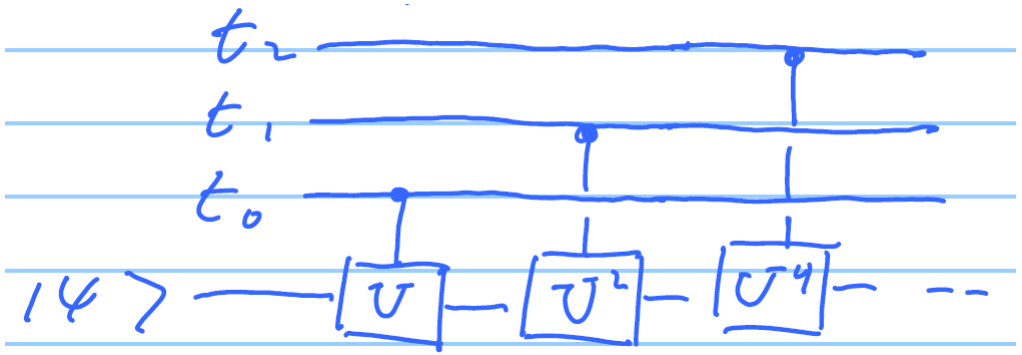
Circuit size ~ (spacetime volume)$^2 \times$ Solovay-Kitaev factor polylog($1/\Delta^2$ h^2)=polylog($h^2(Mt)^2/\delta^2$).

This can be systematically improved using higher-order “Suzuki-Trotter” approximations (or other methods).

Estimating energy eigenvalues

It is hard in general to diagonalize a $2^n \times 2^n$ matrix. But using phase estimation, we can measure eigenvalues, and prepare eigenstates of H if we can simulate time evolution governed by H (as we can if H is local).





Simulate $e^{-iHTt} = U^t$ for $t \in \{1, 2, 4, \ldots, 2^m - 1\}$

$\Rightarrow$ estimate e^{-iET} $\Rightarrow$ estimate $E \bmod 2\pi / T$.

Histogram after many trials. Peak location determines energy, peak height determines overlap of energy eigenstate with $|\psi\rangle$.

Cost estimate

Phase estimation with an *m*-bit ancilla can in principle provide *m* bits of accuracy in our estimate of the energy eigenvalue *E*, but only if our simulation of exp(-i*Ht*) is sufficiently accurate.

It suffices to achieve an error 2^{-m} for evolution over time $t = 2^m$ T. For the geometrically local case:

$$L = O\left(\frac{h^2 (nt)^2}{\delta}\right) = O\left(h^2 (nT)^2 \times \frac{2^{2m}}{2^{-m}}\right) = O\left((hT)^2 n^2 2^{3m}\right)$$

$$\Rightarrow \quad \text{accuracy } 2^{-m} \approx n^{-c} \text{ using a circuit with size } L = O(n^{3c+2}) = \text{poly}(n).$$

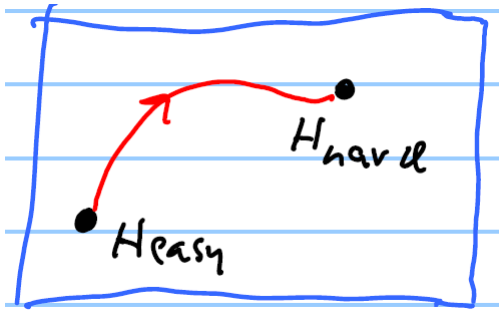
But there is a catch: To estimate an energy eigenvalue (and prepare the corresponding energy eigenstate) to 1/poly(*n*) accuracy in poly(*n*) time, the initial state must have an overlap with the energy eigenstate at least 1/poly(*n*). In some cases, this might be difficult to achieve. (A random state in the *n*-qubit Hilbert space has overlap with any fixed pure state which is exponentially small in *n*.)

Suppose, for example, our goal is to measure the ground state energy. There is a general procedure for preparing ground states which works sometimes (but not always).

This procedure invokes the *quantum adiabatic theorem*: if the Hamiltonian changes sufficiently slowly as the state evolves, then (under suitable conditions) the ground state of the initial Hamiltonian evolves to (a good approximation to) the ground state of the final Hamiltonian.

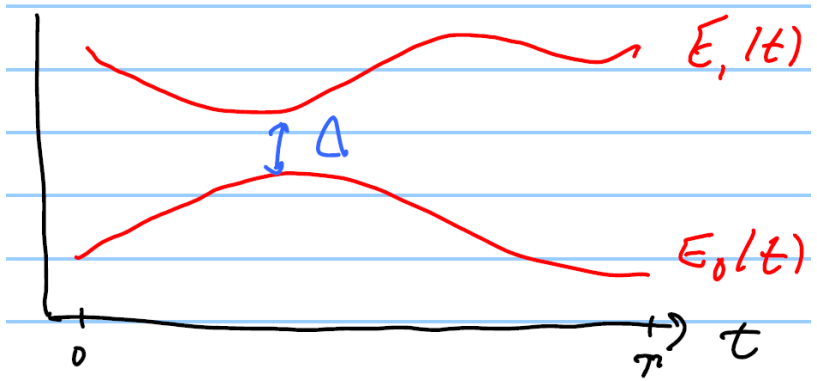
Adiabatic evolution

$H(0) = H_{\text{easy}}, \quad H(T) = H_{\text{hard}}, \quad 0 \le t \le T; \quad$ for example, $\quad H(t) = (1 - t/T)H_{\text{easy}} + (t/T)H_{\text{hard}}.$



$$\Delta = \min_{t \in [0,T]} (E_1(t) - E_0(t)),$$

$$\mathrm{T} > \frac{A}{\Delta^c}, \quad A, c = \text{constant}.$$



It works if the energy gap encountered during the excursion of the Hamiltonian gets no smaller than 1/poly(*n*) --- in that case we have a complete quantum algorithm for estimating the ground state energy to 1/poly(*n*) accuracy in poly(n) time.

But it fails if the energy gap becomes superpolynomially small somewhere along the path followed by the Hamiltonian.

We think it sometimes fails. For some local Hamiltonians, estimating the ground state energy to accuracy 1/poly(*n*) is too hard a problem even for a quantum computer. In fact, the problem is *QMA-hard* --- if we could solve it efficiently, we could solve any problem in QMA (which seems very unlikely.) I'll explain why next time.