Sadly, we now know some things that quantum computers cannot do. But happily, there are also some important things that they can do.

- They can solve the hidden subgroup problem for finitely generated abelian groups, with an exponential speedup (in the oracle model) relative to classical algorithms
- They can speedup, quadratically, exhaustive search for solutions to combinatorial problems

What else can they do?

An important application for quantum computers is simulating the time evolution of quantum systems. The simulation is efficient if the Hamiltonian H is local.

For a system of n qubits, we say that H is "K-local" if

$$H = \sum_a H_a,$$

where each term H_a acts nontrivially on at most K qubits — i.e. $H_a = \tilde{H}_a \otimes I^{n-K}$, and $\tilde{H}_a$ acts on some set of at most K qubits

(of course, we may use a similar definition for a system of d-dimensional subsystems, for constant $d > 2$.) We say that H is local if it is K-local for some constant K.

There is a stronger notion of locality we sometimes use, which can be called "geometrical locality" or "spatial locality." A K-local Hamiltonian is geometrically local in D dimensions if the

qubits can be arranged in (flat) D-dimensional space with a bounded number of qubits per unit volume, and the k-qubits upon which H_a acts nontrivially are all contained in a ball of constant radius. In this sense there are no long-range interactions among the qubits. H is geometrically local if it is geometrically local in D dimensions for some constant D and k.

If we write $H = \sum_a H_a$ where there is a unique H_a for each set of qubits, then the expansion of a k-local H contains at most $\binom{n}{k} = O(n^k)$ terms, and the expansion of a geometrically local H contains $O(n)$ terms (each qubit is contained in a constant number of interacting sets). Let us also assume that each H_a is bounded

$$\|H_a\|_\infty \le h \quad \text{for all } a, \text{ where } h \text{ is a constant}$$

Physicists are interested in geometrically local Hamiltonians because they seem to provide an accurate description of Nature. Therefore it is noteworthy that quantum circuits can simulate quantum evolution governed by a local Hamiltonian efficiently: evolution of n qubits for time t can be simulated to constant accuracy using a circuit whose size is polynomial in n and t.

We can formulate the problem this way: suppose we are given an initial quantum state $|\psi(0)\rangle$, or a classical description of a quantum circuit that prepares the state. Our goal is to construct

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle$$

where $U(t)$ satisfies $\frac{d}{dt}U(t) = -iH(t)U(t)$ and boundary condition $U(0) = I$. (Thus $U(t) = e^{-iHt}$ in

the case where H is time independent. We will settle for computing $|\psi(t)\rangle$ to constant accuracy, i.e. constructing $|\tilde{\psi}(t)\rangle$ where

$$\| |\tilde{\psi}(t)\rangle - |\psi(t)\rangle \| < \delta \quad \text{and} \quad \delta = \text{constant}$$

To relate this to a task that can be described classically, suppose the goal is to sample the prob. distribution

$$\langle \psi(t) | \Pi_a | \psi(t) \rangle \quad \text{where } \Pi_a \text{ projects onto}$$

an eigenstate with eigenvalue a of an observable A that can be measured efficiently by a quantum computer. Classically, this task is believed to be hard in at least some cases, because the unitary matrix $U(t)$ is exponentially large ($2^n \times 2^n$). But quantumly we can do the simulation efficiently, if H is a local Hamiltonian

To simulate continuous time evolution on a classical or quantum computer, we choose a small step size Δ, and approximate evolution for time t by a sequence of t/Δ steps. (If H is actually time dependent, assume Δ is small enough that the change of H during a time interval of width Δ can be neglected.) To attain accuracy

$$\| \tilde{U}(t) - U(t) \|_\infty < \delta,$$

where $\tilde{U}$ is the simulated unitary and U is the ideal unitary, the error per time step should be less than $\delta\Delta/t$.

Suppose $H = \sum_a H_a$

is a sum of M k-local terms, and let's consider the geometrically local case, where $M = O(n)$. We will show below that a single time step can be simulated by a product of M local "gates" (unitary transformations that act on a constant

number of qubits). where each "gate" has an error $O(\Delta^2 h^2)$. Therefore the simulation of evolution for time t uses all together Mt/Δ "gates" where we require

$$\frac{Mt}{\Delta}\Delta^2 h^2 \approx \delta \Rightarrow \Delta = O\left(\frac{\delta}{h^2 Mt}\right)$$

Therefore the total number of "gates" is

$$L = O\left[\frac{h^2 (Mt)^2}{\delta}\right]$$

Furthermore each "gate" can be simulated to accuracy $O(\Delta^2 h^2)$ with a universal gate set using $\text{polylog}\left(\frac{1}{\Delta^2 h^2}\right) = \text{polylog}\left(\frac{h^2 (Mt)^2}{\delta^2}\right)$ gates, according to the Solovay-Kitaev theorem. We conclude that the simulation can be done with a quantum circuit of size

$$L = O\left[\frac{h^2 (Mt)^2}{\delta}\,\text{polylog}\left(\frac{h^2 (Mt)^2}{\delta^2}\right)\right]$$

In the case where H is geometrically local $M = O(n) = O(V)$ where V is the spatial volume of the system. Since h is a constant, we conclude that the cost of simulating time evolution with fixed accuracy scales like

$$L = O(\Omega^2 \,\text{polylog}\,\Omega)$$

where $\Omega = Vt$ is the simulated volume of spacetime

Now we need to explain how to simulate a single time step. We'll use the idea

That $\exp\left(\sum_a A_a\right)$ can be approximated by $\prod_a \exp(A_a)$ if $\|A\| \ll 1$. To check the accuracy, we expand the exponentials:

$$\exp\left(\sum_a A_a\right) - \prod_a \exp(A_a)$$

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

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

$$= \tfrac{1}{2}\left(\sum_{a<b} A_a A_b + \sum_{a>b} A_b A_a\right) - \sum_{a<b} A_a A_b + \cdots$$

$$= -\tfrac{1}{2}\sum_{a<b} [A_a, A_b] + \cdots$$

(where $+\cdots$ denotes terms higher order in A_a)

Writing $H = \sum_a H_a$, then, we find that

$$e^{-iH\Delta} - \prod_a e^{-iH_a\Delta} = \tfrac{1}{2}\Delta^2 \sum_{a<b} [H_a, H_b] + \text{h.o.}$$

Now, how many nonvanishing commutators $\{[H_a, H_b]\}$ can occur in this sum? If there are $M \le \binom{n}{k}$ $H = \sum_a H_a$, and the Hamiltonian is k-local, so that each term acts on k qubits, then the number of nonvanishing commutators is $O(n^{2k-1}) = O(M^{2-1/k})$. For each set of k qubits, there are no more than $k\binom{n}{k-1} = O(n^{k-1})$ sets of k qubits that intersect it. Also, if the Hamiltonian is geometrically local, there are $O(n)$ terms in H, and each term fails to commute with a constant number of terms. So, there are $O(n) = O(M)$ nonvanishing commutators.

We conclude that (in the geometrically local case)

$$\| e^{-iH\Delta} - \prod_a e^{-iH_a\Delta} \| = O(M\Delta^2 h^2)$$

– since $\prod_a e^{-iH_a\Delta}$ is a product of M "gates," we have verified that the accuracy per gate is $O(\Delta^2 h^2)$

(Note that terms arising from the higher order expansion of the exponential are of order $M\Delta^3 h^3$, and therefore systematically expressed by another factor of $\Delta h = O(S/hMt) = O((S/L)^{1/2})$.)

So far we have shown that, for a geometrically local H that is a sum of bounded terms, evolution in a spacetime volume Ω can be achieved with a quantum circuit of size:

$$L = O(\Omega^2 \,\mathrm{polylog}\,\Omega)$$

The resources needed for the simulation scale like the square of the simulated volume (up to a log factor)
Can this be improved?

An improved approximation to $\exp(\sum_a A_a)$ is the subject of HW exercise (6.4). Instead of $\prod_a e^{A_a}$ we use

$$\prod_{a\rightarrow} e^{\frac{1}{2}A_a} \prod_{a\leftarrow} e^{\frac{1}{2}A_a}$$

where $\prod_{a\rightarrow}$ denotes product in ascending order and $\prod_{a\leftarrow}$ denotes product in descending order. The HW shows that, for geometrically local H,

$$\| e^{-i\Delta H} - \prod_{a\rightarrow} e^{-iH_a\Delta/2} \prod_{a\leftarrow} e^{-iH_a\Delta/2} \| = O(M h^3 \Delta^3),$$

i.e. the error per gate is $O(h^3\Delta^3)$ instead of $O(h^2\Delta^2)$

For an accurate simulation, we choose

$$\frac{Mt}{\Delta}(\Delta^3 h^3) \approx S \Rightarrow \Delta \approx \left(\frac{S}{h^3 Mt}\right)^{\frac{1}{2}},$$

the number of "gates" is

$$\frac{Mt}{\Delta} \approx \frac{h^{3/2}(Mt)^{3/2}}{\delta^{1/2}}$$

and the Solovay-Kitaev blowup factor is

$$\text{polylog}\left(\frac{1}{\Delta^3 h^3}\right) = \text{polylog}\left(\frac{h^{3/2}(Mt)^{3/2}}{\delta^{3/2}}\right) = \text{polylog}\left(\frac{hMt}{\delta}\right)$$

We conclude that spacetime volume Ω can be simulated with a circuit of size

$$\mathcal{L} = O(\Omega^{3/2}\,\text{polylog}\,\Omega)$$

(The improved approximation in each time step increases the circuit size by only a factor of 2)

The accuracy of the "Trotter-Suzuki approximation" to $e^{-iH\Delta}$ can be improved further, so that

$$\| e^{-iH\Delta} - \text{approx} \| = O(c_p M (h\Delta)^{p+1})$$

where p is any power, the constant c_p depends on p, and the improved approximation increases the number of "gates" by a factor that depends on p. With this approximation, we choose

$$\frac{Mt}{\Delta}(h\Delta)^{p+1} \approx \delta \Rightarrow \Delta \approx \frac{\delta^{1/p}}{h^{(p+1)/p}(Mt)^{1/p}}$$

so that the number of "gates" is

$$\frac{Mt}{\Delta} \approx \frac{h^{(p+1)/p}(Mt)^{(p+1)/p}}{\delta^{1/p}}$$

Including the Solovay-Kitaev log factor, we can do the simulation with a circuit of size

$$\mathcal{L} = O(\mathcal{R}^{1+\varepsilon} \operatorname{polylog} \mathcal{R})$$

where $\varepsilon = 1/p$ (but the constant factors blow up as $\varepsilon \to 0$). So as we systematically improve the approximation to evolution for a single time step, the circuit size approaches scaling like the simulated volume, but never quite makes it. Somehow, Nature manages to simulate herself with "resources" scaling like $\mathcal{R}$, but we don't know how to do quite as well with a universal quantum computer.

Whether the quantum circuit model can simulate Nature efficiently is an important issue, because it addresses whether this model is the right one for studying what problems can be solved with reasonable resources by computers that are in principle physically realizable. We believe that the classical Turing machine model is not the right model, because it seems to be incapable of simulating general quantum systems, even ones for which the the Hamiltonian is geometrically local. The quantum circuit model is presumably stronger, but is it really strong enough?

Particle physicists model fundamental physics using quantum field theory. Some predictions of QFT can be computed classically, and the success of such predictions provides the evidence that QFT is a good model. But we don't know how to simulate real time evolution of, for example, nuclear matter using quantum chromodynamics on a classical computer. Can we do such simulations on a quantum computer?

The question is subtle because the number of qubits per unit volume is infinite in QFT (there are degrees of freedom at arbitrarily short distances) and although the Hamiltonian is local, the

terms in the Hamiltonian do not necessarily have bounded norm. On the other hand, in a physical process of interest, we can usually assume that the energy density per unit volume is bounded above. In that case, we expect that the very-short-distance degrees of freedom are not so relevant, and that the local Hamiltonian can be well approximated by an operator with bounded norm. A reasonably expectation is that the complexity of a simulation of the dynamics scales polynomially in

$$\Omega \ell_{max} = (V t \ell_{max}) = E_{max} t$$

where ℓ_{max} is the maximal energy per unit volume, and E_{max} is an upper bound on the total energy. But there is no rigorous theorem establishing such scaling. Though physicists have a pretty good grasp of the properties of QFT (as indicated by the satisfying agreement of many predictions with experimental data), rigorous mathematical results pertaining to QFT are still quite technical and incomplete.

Our understanding is even less satisfactory for physical processes in which gravitational effects are important. Can the quantum circuit model simulate quantum gravity efficiently? If so, quantum computers may turn out to be a powerful tool for deepening our understanding of quantum gravity. If not, then we still have not found the computational model that fully captures the computational power that is potentially realizable in Nature.

Estimating energy eigenvalues and preparing energy eigenstates

Ph/CS 219, 2 March 2009

We have argued that a quantum computer can efficiently simulate the time evolution of a quantum system with a local Hamiltonian; i.e., it can solve the *time-dependent* Schroedinger equation. Another thing that physicists and chemists want to do is solve the *time-independent* Schroedinger equation; i.e., compute the energy eigenvalues of a Hamiltonian. For example, chemists say that estimating the ground state energy of a molecule to "chemical accuracy" (about one part in a million) is valuable for predicting the properties of chemical reactions.

[Actually, it is a subtle question whether the Hamiltonians typically studied by quantum chemists can be regarded as *local*. The goal is the determine the quantum state of many electrons with the positively charged nuclei held at fixed positions. The position of the electron is really a continuous quantum variable, but we can express the electron's state in terms of a finite set of orbitals --- the issue is whether the electron orbitals couple with one another only in clusters of constant size as the number of electrons increases.]

In general, finding the energy eigenvalues of a local Hamiltonian seems to be a hard problem classically because we need to diagonalize a $2^n \times 2^n$ matrix for a system of n qubits. In some cases this may be easy, for example if the matrix is very sparse, but in some physically relevant cases efficient classical algorithms are not known.

Sometimes, if we express H in a "natural" basis, we find that all the off-diagonal terms in the matrix H are nonpositive, i.e.

$$H = cI - h$$

where I is the identity and h has only nonnegative entries. In that case, the ground state $|\psi_0\rangle$ of H (the eigenvector with the lowest eigenvalue) can be expressed as $|\psi_0\rangle = \sum_i c_i |i\rangle$

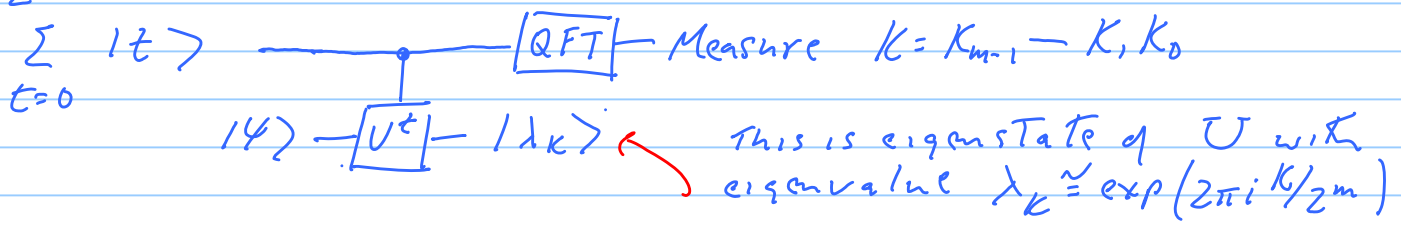
where $|i\rangle$ is a basis element and all c_i can be chosen nonnegative. That is because $|\psi_0\rangle$ maximizes

$$\langle \psi_0 | h | \psi_0 \rangle = \sum c_i^* c_j h_{ij}$$

and so it is optimal to choose $c_i^* c_j \geq 0$ for all i and j. When all the c_i are nonnegative, there are efficient "quantum Monte Carlo" sampling algorithms that can find the $\{c_i\}$ efficiently and accurately. But if the off-diagonal terms in

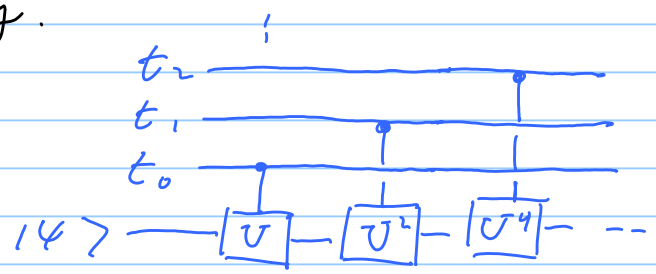
H have both + and − signs, then sampling algorithms might not work well because there can be delicate cancellations between positive and negative terms contributing to $\langle \psi_0 | H | \psi_0 \rangle$. Computational physicists call this the "sign problem."

But --- on a quantum computer, we can estimate eigenvalues of a unitary matrix U using the "phase estimation" algorithm. To obtain m bits of accuracy, we prepare an m-qubit register in a uniform superposition state, and execute this circuit:



This measured value of K provides an estimate of the eigenvalue to m bits of accuracy.

To perform U^t conditioned on t, we simulate e^{-iHt} for



$$t \in T \times \{1, 2, 4, 8, \ldots 2^{m-1}\}$$

This suffices to find the fractional part of $\frac{E T}{2\pi}$ to m-bit accuracy, where E is an eigenvalue of the Hamiltonian H. We choose the step size in the simulation of e^{-iHt} so that the accuracy is $\delta \approx 2^{-m}$ for $t = 2^m T$. If the Hamiltonian is geometrically local, we have seen that the step size should be

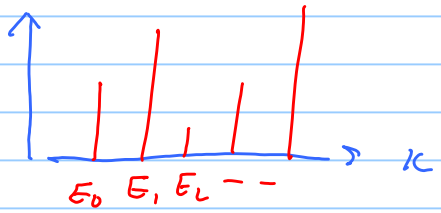
$$\Delta = O\left(\frac{\delta}{h^2 n t}\right) = O\left(\frac{2^{-2m}}{h^2 n T}\right)$$

The circuit size (neglecting the log factor) is

$$L = O\left(\frac{nt}{\Delta}\right) = O\left(n 2^m T \times \frac{h^2 n T}{2^{-2m}}\right)$$

$$= O\left(2^{3m} h^2 (nT)^2\right)$$

For a particular preparation of the state $|\psi\rangle$, suppose we repeat the computation many times, and plot a histogram of the results:



The location of each narrow peak estimates an energy eigenvalue, modulo $2\pi/T$.

The height of the peak estimates $|\langle E_a|\psi\rangle|^2$ – the overlap of $|\psi\rangle$ with the corresponding energy eigenstate.

To compute the energy eigenvalue to accuracy polynomial in n, we choose

$$2^{-m} \sim \frac{1}{n^c} \Rightarrow m = c \log_2 n$$

The algorithm is efficient: the quantum circuit size is

$$O\left(2^{3m} h^2 (nT)^2\right) = O\left(n^{3c} n^2\right)$$

However--- if we want to estimate the ground state energy E_0 to polynomial accuracy in quantum polynomial time, we must be able to prepare a state $|\psi\rangle$ whose overlap with the ground state $|E_0\rangle$ is only polynomially small:

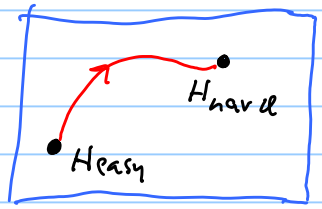
$$|\langle E_0|\psi\rangle|^2 > 1/\text{poly}(n)$$

If that is the case, we can get a good estimate of E_0 in only polynomially many trials. As a bonus,

when we obtain the value E_0 for the measured eigenvalue E_0, then we have projected the state $|\psi\rangle$ onto the ground state $|E_0\rangle$, and therefore we can compute further properties of $|E_0\rangle$, such as the distribution $\text{Prob}(a) = \langle E_0 | \Pi_a | E_0 \rangle$, where Π_a is projector onto eigenspace of an efficiently measurable observable.

But— the overlap of a randomly chosen n-qubit state with $|E_0\rangle$ is exponentially small, so preparing $|\psi\rangle$ with only a polynomially small overlap is not necessarily easy.

One way we might attempt to construct a state with a significant overlap with the ground state is by appealing to the quantum adiabatic theorem: We prepare the ground state of a Hamiltonian H_{easy} whose ground state is easy to find classically. Then we simulate Schrödinger evolution governed by a time-dependent Hamiltonian $H(t)$ such that



$$H(0) = H_{easy} \qquad H(T) = H_{hard}, \qquad 0 \le t \le T$$

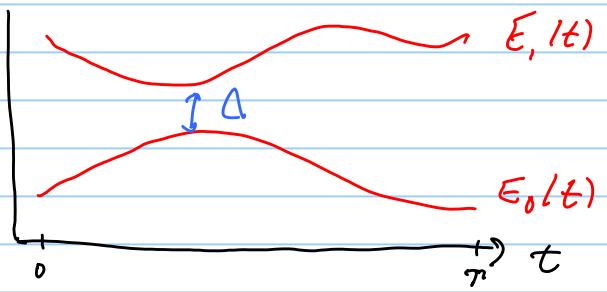
where H_{hard} is the Hamiltonian whose ground state we wish to construct. For example, we might choose

$$H(t) = (1 - t/T) H_{easy} + (t/T) H_{hard}.$$

The quantum adiabatic theorem says that if $|\psi(0)\rangle$ has a large overlap with the ground state of H_{easy}, then $|\psi(T)\rangle$ has a large overlap with the ground state of H_{hard}, <u>if</u> T is long enough (i.e., the Hamiltonian $H(t)$ changes slowly enough). But how slow is slow enough? Let $E_0(t)$ denote the energy of the ground state of $H(t)$ and let $E_1(t)$ denote the energy of the first excited state of

$H(t)$. Define $\Delta = \min_{t \in [0,\pi]} \left(E_1(t) - E_0(t) \right)$;

we say Δ is the minimum "energy gap" between the ground and first excited state. Then the adiabatic theorem says that



$$T > \frac{A}{\Delta^c}$$ is slow

enough, where A and c are constants independent of the number of qubits n. Therefore --- we have a complete polynomial algorithm for computing the ground state energy of a local Hamiltonian to polynomial accuracy in polynomial time provided that

$$\Delta > \frac{1}{\text{poly}(n)}.$$

But if Δ becomes exponentially small in n during the evolution in which $H(t)$ interpolates between H_{easy} and H_{hard}, then this algorithm requires exponential time.

Unfortunately, we'll see that it follows from weak complexity-theoretic assumptions that there are local Hamiltonians for which computing ground state energy is hard even for a quantum computer (for example, otherwise we would be able to solve efficiently any problem in NP using a quantum computer, which seems unlikely). So the energy gap must be smaller than polynomially small in such cases.

On the other hand -- it is reasonable to be hopeful that computing ground state energy for quantum systems will be an important application of quantum computing (e.g. see A. Aspuru-Guzik et al., Science 309, 1704 (2005)). Chemists say it is valuable to compute the energy of the electronic ground state of a molecule with atomic nuclei at fixed positions to accuracy

$\sim 10^{-6}$ ("chemical accuracy"). They claim that it is an adequate approximation to express

$$H = \sum_{a=1}^{M} H_a \quad \text{where} \quad M \cong (\#\text{of atoms})^4$$

and each H_a acts on ≤ 4 basis functions. Hence, this Hamiltonian is local. They estimate that $n \cong 47$ qubits suffice to compute H_2O ground state energy to chemical accuracy, which is competitive with the best current classical computations. For caffeine (21 atoms), about 300 qubits should suffice. Furthermore, chemists assert (without proof) that it is possible to evolve adiabatically from the "Hartree-Fock" Hamiltonian (which they can solve) to the "full configuration interaction" (FCI) Hamiltonian (which they want to solve) while the gap satisfies $\Delta > \text{constant}$. If that is right, quantum computers are likely to be a powerful tool for studying molecular chemistry.

How hard is it to simulate a quantum computer?

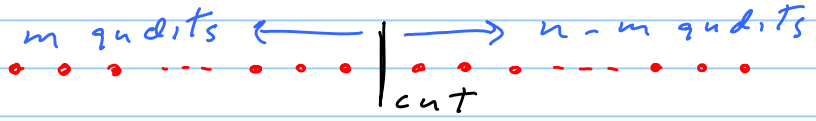
Clearly, we would like to understand more deeply the classical and quantum complexity of solving the time-dependent and time-independent Schroedinger equations. In particular, we wish to identify cases for which the problem seems to be hard classically and easy quantumly, for these are cases where quantum computers will find a useful niche.

More broadly, why do we believe that quantum computers are more powerful than classical ones, and what is the source of the quantum computer's power? Roughly speaking, it seems to be hard to simulate a quantum system with a classical computer because the Hilbert space of the quantum computer is exponentially large in the number of qubits *n*, and that exponentially large Hilbert space is needed to accommodate and describe the quantum correlations among the qubits in a many-body quantum system. From this perspective, it seems legitimate to claim that quantum entanglement is the source of the quantum computer's power.

On the other hand, that claim may be too simplistic, because:
1) Some quantum computations that generate highly entangled quantum states are easy to simulate classically. We'll see an example next term when we discuss (in connection with quantum error correction) quantum computation using the Clifford group.

2) For *mixed* states, simulating a quantum computation classically might be hard even if the state remains separable (that is, unentangled) at all times during the computation --- even if the correlations among the parts of the quantum computer are "classical" they could still be hard to simulate. You examined an example for HW problem (6.1): estimating the trace of an exponentially large unitary matrix using just "one clean qubit".

So let's ask the question this way: for for quantum computation with pure states, if the qubits in the computer never becomes highly entangled during the course of the computation, can we simulate the quantum computer efficiently with a classical computer? As we'll see, the answer is yes.



Imagine that n qudits (d-dimensional systems) are arranged in a line, and consider cutting the systems into two parts anywhere along the line: there are m qudits to the left of the cut and n-m qubits to the right of the cut. Suppose that for any way of choosing where we cut the chain, the entanglement between the two parts is bounded above by a constant (independent of the system size n). We could quantify the entanglement in various ways, and the conclusion would be the same, but to keep the discussion simple let us use the Schmidt number. Recall that the Schmidt number is the number of terms in the Schmidt expansion of a bipartite pure state --- equivalently it is the rank of the density operator for either part.

while in principle the Schmidt number could be as large as $d^{n/2}$ when the system is cut into two subsystems of equal size, we assume

$$\text{Schmidt Number} \leq D = \text{constant}$$

We claim that under this assumption:

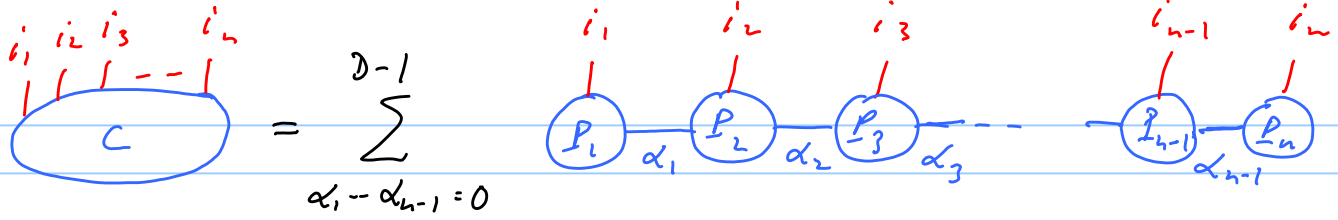
(1) there is a succinct classical description of the pure state of n qudits

(2) A computation in which the n qudits have bounded Schmidt Number at all times can be efficiently simulated on a classical computer

First, let's construct the succinct description. The n-qudit state $|\psi\rangle$ can be expanded in the standard basis as

$$|\psi\rangle = \sum C^{i_1 i_2 \cdots i_n} |i_1, i_2 \cdots i_n\rangle$$

in terms of the d^n complex numbers $\{C^{i_1 \cdots i_n}\}$. This is not succinct in general, but if the Schmidt rank is bounded then each $C^{i_1 \cdots i_n}$ can be expressed as a contraction of tensors, where each tensor has 3 indices, each with a constant number of values:



Here each $(P_k)^{i_k}_{\alpha_{k-1}\alpha_k}$ has $\leq D^2$ entries (except for the sites at the ends of the chain, which have $\leq D$ entries). Therefore, the state is described by $\leq n D^2$ complex parameters, a number that is linear rather than exponential in n. (In the case of a product state, this becomes nd parameters.)

To obtain this decomposition of the n-site state $|\psi\rangle$ we perform the Schmidt decomposition repeatedly ($n-1$ times in succession). In the first step we divide the system between qudits 1 and 2:

$$|\psi\rangle = \sum_{\alpha_1} \lambda_{1\alpha_1} |\psi_{1\alpha_1}\rangle |\phi_{2\alpha_1}\rangle.$$

Here $\lambda_{1\alpha_1}$ is a Schmidt coefficient $|\psi_{1\alpha_1}\rangle$ is an element of an ON basis for the first qudit, and $|\phi_{2\alpha_1}\rangle$ is a state of qudits $2,3,\ldots,n$. We can expand $|\psi_{1\alpha_1}\rangle$ in the standard basis, obtaining

$$|\psi\rangle = \sum_{\alpha_1} \sum_{i_1} \lambda_{1\alpha_1} \psi^{i_1}_{1\alpha_1} |i_1\rangle |\phi_{2,\alpha_1}\rangle$$

$$= \sum_{\alpha_1} (P_1)^{i_1}_{\alpha_1} |i_1\rangle |\phi_{2,\alpha_1}\rangle \quad \text{where } (P_1)^{i_1}_{\alpha_1} = \lambda_{1\alpha_1} \psi^{i_1}_{1\alpha_1}$$

Now we Schmidt decompose $|\phi_{2,\alpha_1}\rangle$ across the cut between qudits 2 and 3, finding

$$|\phi_{2,\alpha_1}\rangle = \sum_{\alpha_2} \sum_{i_2} (P_2)^{i_2}_{\alpha_1\alpha_2} |i_2\rangle |\phi_{3,\alpha_2}\rangle$$

Note that there is no α_1 label on the states $\{|\psi_{3,\alpha_2}\rangle\}$ that is because these are the states occuring in the Schmidt decomposition across the 2-3 cut that describe the right side of the cut. These states have nothing to do with α_1, which labels the states in the Schmidt decomposition across the 1-2 cut.

Repeating the Schmidt decomposition $n-1$ times we find

$$|\psi\rangle = \sum_{\alpha_1 \cdots \alpha_{n-1}} \sum_{i_1 \cdots i_n} (\underline{P}_1)^{i_1}_{\alpha_1} (\underline{P}_2)^{i_2}_{\alpha_1\alpha_2} (\underline{P}_3)^{i_3}_{\alpha_2\alpha_3} \cdots (\underline{P}_n)^{i_n}_{\alpha_{n-1}} \times |i_1 i_2 \cdots i_n\rangle$$

This is the decomposition in terms of contracted tensors that we sought. It is called the "matrix-product-state" (MPS) decomposition of $|\psi\rangle$. It exists for any $|\psi\rangle$, but for generic $|\psi\rangle$, the matrices in the middle of the chain become exponentially large. Under the assumption of bounded Schmidt rank, however,

$\underline{P}_1^{i_1}$ and $\underline{P}_n^{i_n}$ are D-component vectors

$(\underline{P}_k)^{i_k}$ for $k = 2, 3, \ldots, n-1$ are $D \times D$ matrices.

Therefore $|\psi\rangle = \sum_{i_1 \cdots i_n} \underline{P}_1^{i_1} \underline{P}_2^{i_2} \cdots \underline{P}_n^{i_n} |i_1 i_2 \cdots i_n\rangle$

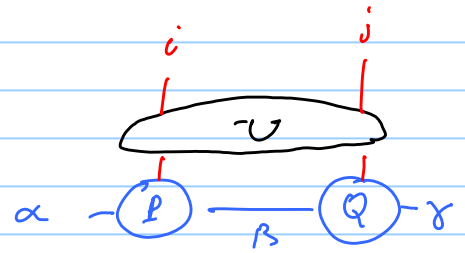
← where this is a product of matrices

If we don't want the endpoints to be special we can replace $\underline{P}_1^{i_1}$ and $\underline{P}_n^{i_n}$ by $D \times D$ matrices contracted with one another -- then

$$|\psi\rangle = \sum_{i_1 \cdots i_n} \mathrm{Tr}(\underline{P}_1^{i_1} \underline{P}_2^{i_2} \cdots \underline{P}_n^{i_n}) |i_1 i_2 \cdots i_n\rangle$$

This description looks like the MPS locally, but with the ends of the chain now glued together.

Next we want to see that the MPS description can be efficiently updated when local quantum gates act on the state. Suppose a unitary transformation U acts on a pair of neighboring qudits on the chain



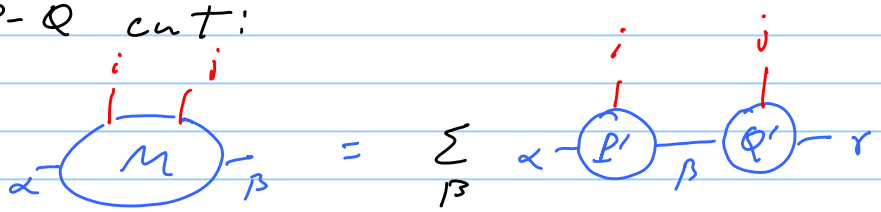
The unitary affects only two neighboring matrices:

$$\sum_\beta (P)^i_{\alpha\beta} (Q)^j_{\beta\gamma}$$

$$\longmapsto \sum_\beta \sum_{i'j'} U^{ij}_{i'j'} (P)^{i'}_{\alpha\beta} (Q)^{j'}_{\beta\gamma}$$

$$\equiv M^{ij}_{\alpha\gamma}$$

Now we need to update the Schmidt decomposition between the P-Q cut:



We regard $M^{ij}_{\alpha\gamma}$ as a $dD \times dD$ matrix, and we perform its singular value decomposition (SVD). Recall that for any matrix M, there are unitary matrices V_L V_R such that $M = V_L^\dagger \Delta V_R$. This Schmidt decomposes the state

$$\sum_{ac} M_{ac} |ac\rangle = \sum_b \Delta_b |\psi_{Lb}\rangle \otimes |\psi_{Rb}\rangle \quad \text{where} \quad |\psi_{Lb}\rangle = \sum_a |a\rangle (V_L^\dagger)_{ab}$$

$$|\psi_{Rb}\rangle = \sum_c |c\rangle V_{Rbc}$$

Here $|ac\rangle$ is shorthand for $|\alpha i j \gamma\rangle$, where $|\alpha\rangle$ labels the states in the Schmidt decomp. across the cut to the left of i and γ labels states in the Schmidt decomp. across the cut to the right of j.

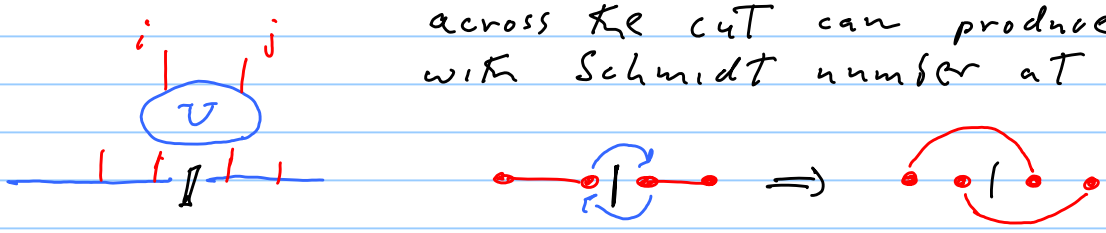
The SVD of an $N \times N$ matrix can be computed in time $O(N^3)$ on a classical computer. In the case of $M^{ij}_{\alpha\gamma}$, for which $N = dD$, this is time $O(d^3 D^3)$. In general, the middle index b would be summed over N ($= dD$) values. But if the Schmidt rank stays bounded by D, the matrix Δ has at most D nonzero eigenvalues. We may identify

$$(P')^{i}_{\alpha\beta} = \left(V_L^{+}\sqrt{\Delta}\right)_{\alpha\beta} \quad \text{and} \quad (Q')^{j}_{\beta\gamma} = \left(\sqrt{\Delta}\, V_R\right)_{\beta\gamma}$$

If we want to simulate a 2-qudit gate between two qudits that are not neighbors, we can perform a sequence of (at most n) SWAP gates to bring the qudits in neighboring positions, perform U, and then SWAP back. A SWAP gate acting across a cut might increase the Schmidt number, but by at most a factor of d^2 (see below). So in each step the Schmidt number is at most $d^2 D$, and the SVD can be computed in time $O(d^9 D^3)$

the gate acting on an arbitrary pair of qudits can be simulated in $O(n d^9 D^3)$ steps on a classical computer. Since d and D are constants, a quantum circuit with T gates can be simulated with $O(n T)$ classical gates.

Acting on a product state, a 2-qudit state acting across the cut can produce a state with Schmidt number at most d^2



This can be achieved by a SWAP gate acting on two max. entangled pair, one on each side of the cut. Therefore it can increase the Schmidt number D to at most $d^2 D$. This means that, starting with a product state, a circuit in which no more than G gates act across any particular cut creates a state with Schmidt number at most $D = (d^2)^G$ across that cut. (We can simulate a 2-qudit gate acting across the cut by SWAPing the qudits until they are at adjacent sites on either side of the cut, applying U, then swapping back. Only U, not the swaps, increases the Schmidt number, and it increases it by at most the factor d^2.)

The quantum circuit can be efficiently simulated if $D = \text{poly}(n)$ across every cut, and thus if $G = O(\log n)$ across each cut

We should also note that for an MPS, local measurements of the qudits in the chain are easy to simulate. First let's impose the proper normalization condition on the state $|\psi\rangle$:

$$|\psi\rangle = \sum_{i_1 \cdots i_n} \mathrm{tr}\left(P_1^{i_1} \cdots P_n^{i_n}\right) |i_1, i_2, \ldots, i_n\rangle$$

$$\Rightarrow \quad \langle\psi|\psi\rangle = \sum_{i_1 \cdots i_n} \left|\mathrm{tr}\left(P_1^{i_1} \cdots P_n^{i_n}\right)\right|^2, \text{ which is}$$

conveniently written as

$$\sum_{i_1 \cdots i_n} \mathrm{tr}\left(P_1^{i_1} \otimes P_1^{i_1 *}\right)\left(P_2^{i_2} \otimes P_2^{i_2 *}\right) \cdots \left(P_n^{i_n} \otimes P_n^{i_n *}\right)$$

$$= \mathrm{tr}\left(E_1 E_2 \cdots E_n\right) \quad \text{where } E_k = \sum_{i_k} P_k^{i_k} \otimes P_k^{i_k *}$$

is a $D^2 \times D^2$ matrix

The expectation of a local observable can be expressed as

$$\langle\psi|\, \sigma_1 \otimes \sigma_2 \otimes \cdots \otimes \sigma_n |\psi\rangle$$

$$= \sum_{j_1 \cdots j_n} \sum_{i_1 \cdots i_n} \mathrm{tr}\left(P_1^{i_1 *} \cdots P_n^{i_n *}\right) \langle j_1|\sigma_1|i_1\rangle \cdots \langle j_n|\sigma_n|i_n\rangle \times \mathrm{tr}\left(P_1^{i_1} \cdots P_n^{i_n}\right)$$

$$= \mathrm{tr}\left(E_1(\sigma_1)\, E_2(\sigma_2) \cdots E_n(\sigma_n)\right)$$

where $$E_k(\sigma_k) = \sum_{i_k j_k} \left(P_k^{i_k} \otimes P_k^{j_k *}\right) \langle j_k|\sigma_k|i_k\rangle$$

is also a $D^2 \times D^2$ matrix. We can evaluate the trace of a product of n constant-size matrices in time $O(n)$. This completes the proof that slightly entangled quantum computations can be classically simulated.

Finally, let's return to the problem of finding the energy of the ground state of a local Hamiltonian. In many one-dimensional systems studied in

physics, the ground state and low lying excited states can be well approximated by an MPS with reasonable matrix size. That is, there is a good approximation to e.g. the ground state with maximal Schmidt number across any cut

$$D = \text{poly}(n)$$

Since the state has a succint (poly in n) representation, the ground state energy E_0 can be computed by an efficient variational procedure. However, there are local Hamiltonians (even translation-invariant ones) for which finding E_0 seems to be hard not only for classical computers but also for quantum computers! In these cases, an accurate MPS representation apparently requires matrices of superpolynomial size.