Quantum algorithms (CO 781, Winter 2008)
Prof. Andrew Childs, University of Waterloo

LECTURE 12: Simulating Hamiltonian dynamics

So far, we have described quantum algorithms using the most common model of quantum computation, the quantum circuit model. In this model, an algorithm corresponds to a sequence of local (one- and two-qubit) unitary gates, and complexity is measured by the total number of gates. However, we defined a continuous-time quantum walk as the solution of a differential equation, the Schrödinger equation. Even when the Hamiltonian acts locally on a graph, this will generally give a unitary operator that acts nontrivially on all the qubits used to represent the vertices. In this lecture we will see consider simulating such dynamics in the quantum circuit model. This gives us a way to quantify the complexity of the dynamics generated by a particular Hamiltonian.

Of course, real quantum systems naturally evolve continuously in time according to some Hamiltonian, so we could also imagine defining a notion of efficient computation directly in terms of the Hamiltonian. However, it is often more convenient to work with the circuit model, especially in the context of query complexity.

Efficient simulation We will say that a Hamiltonian H acting on n qubits can be *efficiently simulated* if for any $t > 0$, $\epsilon > 0$ there is a quantum circuit U consisting of $\mathrm{poly}(n, t, 1/\epsilon)$ gates such that $\|U - e^{-iHt}\| < \epsilon$. We would like to understand the conditions under which a Hamiltonian can be efficiently simulated. Our strategy will be to start from simple Hamiltonians that can be easily simulated, and define ways of combining the known simulations to give more complicated ones.

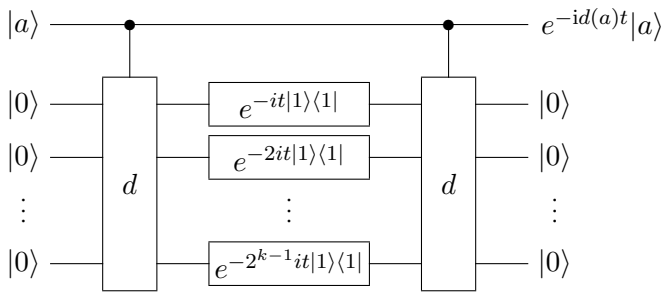
There are a few cases where a Hamiltonian can obviously simulated efficiently. For example, this is the case if H only acts nontrivially on a constant number of qubits, simply because any unitary evolution on a constant number of qubits can be approximated using a constant number of one- and two-qubit gates.

Note that since we require a simulation for an arbitrary time t (with $\mathrm{poly}(t)$ gates), we can rescale the evolution by any polynomial factor: if H can be efficiently simulated, then so can cH for any $c = \mathrm{poly}(n)$. This holds even if $c < 0$, since any efficient simulation is expressed in terms of quantum gates, and can simply be run in reverse.

In addition, we can rotate the basis in which a Hamiltonian is applied using any unitary transformation with an efficient decomposition into basic gates. In other words, if H can be efficiently simulated and the unitary transformation U can be efficiently implemented, then $UHU^\dagger$ can be efficiently simulated. This follows from the simple identity

$$e^{-iUHU^\dagger t} = Ue^{-iHt}U^\dagger. \tag{1}$$

Another simple but useful trick for simulating Hamiltonians is the following. Suppose H is diagonal in the computational basis, and any diagonal element $d(a) = \langle a|H|a\rangle$ can be computed efficiently. Suppose for simplicity that the diagonal element $d(a)$ is expressed as a binary number with k bits of precision. Then H can be simulated efficiently using the following circuit:



For any input computational basis state $|a\rangle$, together with a k-qubit ancilla state initialized to $|0\rangle$, this circuit perform the sequence of operations

$$\begin{aligned} |a,0\rangle &\mapsto |a,d(a)\rangle && (2)\\ &\mapsto e^{-itd(a)}|a,d(a)\rangle && (3)\\ &\mapsto e^{-itd(a)}|a,0\rangle && (4)\\ &= e^{-iHt}|a\rangle|0\rangle. && (5) \end{aligned}$$

So by linearity, the circuit procedure simulates H for time t on an arbitrary input.

Note that if we combine this simulation with the previous one, we have a way to simulate any Hamiltonian that can be efficiently diagonalized, and whose eigenvalues can be efficiently computed.

Adding Hamiltonians Given two or more simulable Hamiltonians, we can produce further simulable Hamiltonians from them. In particular, if H_1 and H_2 can be efficiently simulated, then $H_1 + H_2$ can also be efficiently simulated. If the two Hamiltonians commute, then this is trivial, since $e^{-iH_1t}e^{-iH_2t} = e^{-i(H_1+H_2)t}$. However, in the general case where the two Hamiltonians do not commute, we can still simulate their sum as a consequence of the Lie product formula

$$e^{-i(H_1+H_2)t} = \lim_{m\to\infty}\left(e^{-iH_1t/m}e^{-iH_2t/m}\right)^m. \tag{6}$$

A simulation using a finite number of steps can be achieved by truncating this expression to a finite number of terms, which introduces some amount of error that must be kept small. In particular, if we want to have

$$\left\|\left(e^{-iH_1t/m}e^{-iH_2t/m}\right)^m - e^{-i(H_1+H_2)t}\right\| \leq \epsilon, \tag{7}$$

it suffices to take $m = O((\nu t)^2/\epsilon)$, where $\nu := \max\{\|H_1\|, \|H_2\|\}$. (The requirement that H_1 and H_2 be efficiently simulable means that ν can be at most $\mathrm{poly}(n)$.)

It is somewhat unappealing that to simulate an evolution for time t, we need a number of steps proportional to t^2. Fortunately, the situation can be improved if we use higher-order approximations of (6). For example, one can show that

$$\left\|\left(e^{-iH_1t/2m}e^{-iH_2t/m}e^{-iH_1t/2m}\right)^m - e^{-i(H_1+H_2)t}\right\| \leq \epsilon \tag{8}$$

with a smaller value of m. In fact, by using even higher-order approximations, it is possible to show that $H_1 + H_2$ can be simulated for time t with only $O(t^{1+\delta})$, for any fixed $\delta > 0$, no matter how small.

A Hamiltonian that is a sum of polynomially many terms can be efficiently simulated by composing the simulation of two terms, or by directly using an approximation to the identity

$$e^{-i(H_1+\cdots+H_k)t} = \lim_{m\to\infty} \left(e^{-iH_1 t/m} \cdots e^{-iH_k t/m}\right)^m. \tag{9}$$

Another way of combining Hamiltonians comes from commutation: if H_1 and H_2 can be efficiently simulated, then $i[H_1, H_2]$ can be efficiently simulated. This rule is a consequence of the identity

$$e^{[H_1,H_2]t} = \lim_{m\to\infty} \left(e^{-iH_1\sqrt{t/m}} e^{-iH_2\sqrt{t/m}} e^{iH_1\sqrt{t/m}} e^{iH_2\sqrt{t/m}}\right)^m, \tag{10}$$

which can again be approximated with a finite number of terms. However, I don't know of any situation in which such a simulation is useful.

Sparse Hamiltonians We will say that an $N \times N$ Hermitian matrix is *sparse* (in a fixed basis) if, in any fixed row, there are only $\mathrm{poly}(\log N)$ nonzero entries. The simulation techniques described above allow us to efficiently simulate sparse Hamiltonians. More precisely, suppose that for any a, we can efficiently determine all of the b's for which $\langle a|H|b\rangle$ is nonzero, as well as the values of the corresponding matrix elements; then H can be efficiently simulated. In particular, this gives an efficient simulation of the quantum walk on any graph whose maximum degree is $\mathrm{poly}(\log |V|)$.

The basic idea of the simulation is to simulate the edges in the graph for each color separately, and to combine these pieces using (6). The main new technical ingredient in the simulation is a means of coloring the edges of the graph of nonzero matrix elements of H. A classic result in graph theory (Vizing's Theorem) says that a graph of maximum degree d has an edge coloring with at most $d+1$ colors (in fact, the edge chromatic number is either d or $d+1$). If we are willing to accept a polynomial overhead in the number of colors used, then we can actually find an edge coloring using only local information about the graph.

Lemma. *Suppose we are given an undirected graph G with N vertices and maximum degree d, and that we can efficiently compute the neighbors of any given vertex. Then there is an efficiently computable function $c(a,b) = c(b,a)$ taking $O(d^2 \log^2 N)$ values such that for all a, $c(a,b) = c(a,b')$ implies $b = b'$. In other words, $c(a,b)$ is a coloring of G.*

Proof. Number the vertices of G from 1 through N. For any vertex a, let $\mathrm{idx}(a,b)$ denote the index of vertex b in the list of neighbors of a. Also, let $k(a,b)$ be the smallest k such that $a \neq b \pmod k$. Note that $k(a,b) = k(b,a)$, and $k = O(\log N)$.

For $a < b$, define the color of the edge ab to be the 4-tuple

$$c(a,b) := \big(\mathrm{idx}(a,b), \mathrm{idx}(b,a), k(a,b), b \bmod k(a,b)\big). \tag{11}$$

For $a > b$, define $c(a,b) := c(b,a)$.

Now suppose $c(a,b) = c(a,b')$. There are four possible cases:

1. Suppose $a < b$ and $a < b'$. Then the first component of c shows that $\mathrm{idx}(a,b) = \mathrm{idx}(a,b')$, which implies $b = b'$.

2. Suppose $a > b$ and $a > b'$. Then the second component of c shows that $\mathrm{idx}(a,b) = \mathrm{idx}(a,b')$, which implies $b = b'$.

3. Suppose $a < b$ and $a > b'$. Then from the third and fourth components of c, $k(a,b) = k(a,b')$ and $a = b \pmod{k(a,b)}$, which is a contradiction.

4. Suppose $a > b$ and $a < b'$. Then from the third and fourth components of c, $k(a,b) = k(a,b')$ and $a = b' \pmod{k(a,b')}$, which is a contradiction.

Each case that does not lead to a contradiction gives rise to a valid coloring, which completes the proof. □

Now the simulation proceeds as follows. Write H as a diagonal matrix plus a matrix with zeros on the diagonal. We have already shown how to simulate the diagonal part, so we can assume H has zeros on the diagonal without loss of generality.

Let G be the graph of nonzero matrix elements of H. The vertices of this graph consist of all the computational basis states, and two vertices have an edge between them if they are connected by a nonzero matrix element of H. Use the above lemma to color the edges of this graph, and let $v_c(a)$ be the vertex connected to a by an edge of color c (if there is no such vertex, it does not matter how $v_c(a)$ is defined). Also, let

$$x_c(a) := \mathrm{Re}\,\langle a|H|v_c(a)\rangle \tag{12}$$
$$y_c(a) := \mathrm{Im}\,\langle a|H|v_c(a)\rangle \tag{13}$$

when the vertex a has an incident edge of color c; otherwise, let $x_c(a) = y_c(a) = 0$.

Consider the state space $|a,b,z\rangle$, where the space on which H acts corresponds to states of the form $|a,0,0\rangle$. By assumption, we can efficiently implement unitary operators V_c, W_c defined by

$$V_c|a,b,z\rangle := |a, b \oplus v_c(a), z \oplus x_c(a)\rangle \tag{14}$$
$$W_c|a,b,z\rangle := |a, b \oplus v_c(a), z \oplus y_c(a)\rangle, \tag{15}$$

where $\oplus$ denotes bitwise addition modulo 2. Furthermore, we can efficiently simulate the Hamiltonians S, T where

$$S|a,b,x\rangle := x|b,a,x\rangle \tag{16}$$
$$T|a,b,y\rangle := iy|b,a,-y\rangle. \tag{17}$$

since S, T are easily diagonalized (as they only act nontrivially on two-dimensional subspace). Therefore, we can efficiently simulate the Hamiltonian

$$\tilde{H} := \sum_c (V_c^\dagger S V_c + W_c^\dagger T W_c). \tag{18}$$

When restricted to the subspace of states of the form $|a,0,0\rangle$, we claim that $\tilde{H}$ acts as H:

$$\tilde{H}|a,0,0\rangle = \sum_c [V_c^\dagger S|a, v_c(a), x_c(a)\rangle + W_c^\dagger S|a, v_c(a), y_c(a)\rangle] \tag{19}$$
$$= \sum_c [x_c(a) V_c^\dagger |v_c(a), a, x_c(a)\rangle + i y_c(a) W_c^\dagger |v_c(a), a, -y_c(a)\rangle] \tag{20}$$
$$= \sum_c [x_c(a) + i y_c(a)]|v_c(a), 0, 0\rangle \tag{21}$$
$$= H|a\rangle|0,0\rangle, \tag{22}$$

where in the third line we have used the fact that $v_c(v_c(a)) = a$ when a has an incident edge of color c, and that $x_c(a) = y_c(a) = 0$ otherwise. This shows that H can be efficiently simulated.