7 Hamiltonian simulation

One of the earliest – and most important – applications of a quantum computer is likely to be the simulation of quantum mechanical systems. There are quantum systems for which no efficient classical simulation is known, but which we can simulate on a universal quantum computer. What does it mean to "simulate" a physical system? According to the OED, simulation is "the technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus". What we will take simulation to mean here is approximating the *dynamics* of a physical system. Rather than tailoring our simulator to simulate only one type of physical system (which is sometimes called *analogue* simulation), we seek a general simulation algorithm which can simulate many different types of system (sometimes called *digital* simulation).

According to quantum mechanics, physical systems are specified by *Hamiltonians*. For the purposes of this unit, a Hamiltonian H is a Hermitian operator acting on n qubits (i.e. $H = H^\dagger$), which corresponds physically to a system made up of n 2-level subsystems. The time evolution of the state $|\psi\rangle$ of a quantum system is governed by Schrödinger's equation:

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle,$$

where $H(t)$ is the Hamiltonian of the system (for convenience, we will henceforth absorb $\hbar$ into $H(t)$). An important special case on which we will focus is the *time-independent* setting where $H(t) = H$ is constant. In this case the solution of this equation is

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle,$$

where (as usual) the exponential e^M for an operator M is defined by

$$e^M = \sum_{k \geq 0} \frac{M^k}{k!} = I + M + \frac{M^2}{2} + \frac{M^3}{6} + \ldots.$$

Given a physical system specified by some Hamiltonian H, we would like to simulate the evolution of the system on an arbitrary initial state for a certain amount of time t. In other words, given H, we would like to implement a unitary operator which approximates

$$U(t) = e^{-iHt}.$$

What does it mean to approximate a unitary operator? The "gold standard" of approximation is approximation in the operator norm (aka spectral norm)

$$\|A\| := \max_{|\psi\rangle \neq 0} \frac{\|A|\psi\rangle\|}{\||\psi\rangle\|},$$

where $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$ is the usual Euclidean norm of $|\psi\rangle$. Note that this is indeed a norm, and in particular satisfies the triangle inequality $\|A + B\| \leq \|A\| + \|B\|$. We say that $\widetilde{U}$ approximates U to within ϵ if

$$\|\widetilde{U} - U\| \leq \epsilon.$$

This is a natural definition of approximation because it implies that, for any state $|\psi\rangle$, $\widetilde{U}|\psi\rangle$ and $U|\psi\rangle$ are only distance at most ϵ apart.

7.1 Simulation of weighted sums of Pauli matrices

In order for our quantum simulation of a Hamiltonian H to be efficient, we need $U = e^{-iHt}$ to be approximable by a quantum circuit containing poly(n) gates. A fairly straightforward counting argument shows that not all Hamiltonians H can be simulated efficiently[1]. However, it turns out that several important physically motivated classes can indeed be simulated. Here we will show this for any Hamiltonian that is a sum of polynomially many Pauli matrices on n qubits.

For $s \in \{I, X, Y, Z\}^n$, let σ_s denote the matrix which is a tensor product of the corresponding Pauli matrices, $\sigma_s = s_1 \otimes s_2 \otimes \cdots \otimes s_n$. For example, $\sigma_{ZX} = Z \otimes X$. You will show in the exercises that any Hermitian $2^n \times 2^n$ matrix H can be written as

$$H = \sum_{s \in \{I,X,Y,Z\}^n} \alpha_s \sigma_s$$

for some coefficients $\alpha_s \in \mathbb{R}$. Assuming that only $m = \text{poly}(n)$ of these coefficients are nonzero, we will show that there is an efficient quantum algorithm for simulating H.

An important class of Hamiltonians encapsulated in this class is *k-local* Hamiltonians. A Hamiltonian H of n qubits is said to be k-local if it can be written as a sum

$$H = \sum_{j=1}^{m} H_j$$

for some m, where each H_j is a Hermitian matrix which acts non-trivially on at most k qubits. That is, H_j is the tensor product of a matrix H'_j on k qubits, and the identity matrix on the remaining $n - k$ qubits. For example, the Hamiltonian H on 3 qubits defined by

$$H = X \otimes I \otimes I - 2I \otimes Z \otimes Y$$

is 2-local. Note that, if H is k-local for $k = O(1)$, we can express H as a sum of at most $4^k \binom{n}{k} = \text{poly}(n)$ Pauli matrices. Many interesting physical systems are k-local for small k (say $k \leq 3$), some of which you may be familiar with. Simple examples include the two-dimensional Ising model on a $n \times n$ square lattice,

$$H = J \sum_{i,j=1}^{n} Z^{(i,j)} Z^{(i,j+1)} + Z^{(i,j)} Z^{(i+1,j)}$$

and the Heisenberg model on a line,

$$H = \sum_{i=1}^{n} J_x X^{(i)} X^{(i+1)} + J_y Y^{(i)} Y^{(i+1)} + J_z Z^{(i)} Z^{(i+1)},$$

both of which are used in the study of magnetism and many other areas of physics (in the equations above, $M^{(j)}$ and $M^{(i,j)}$ denotes single-qubit operators acting on the qubit indexed by j or (i, j), respectively, and J, J_x, J_y, J_z are constants).

We first consider the very simple case where H is proportional to a Pauli matrix on n qubits, $H = \alpha s_1 \otimes s_2 \otimes \cdots \otimes s_n$. Then we want to implement the operation

$$e^{-itH} = e^{-it\alpha s_1 \otimes s_2 \otimes \cdots \otimes s_n}.$$

[1] As H is a $2^n \times 2^n$ matrix, it contains exponentially many parameters, whereas a circuit of poly(n) gates only has polynomially many parameters.

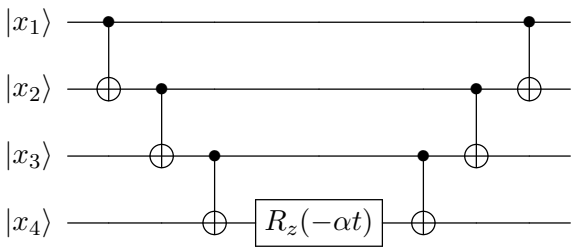
Using the identity $Ue^{-i\alpha H}U^\dagger = e^{-i\alpha UHU^\dagger}$ for any H, which can be shown directly from the definition of the matrix exponential, we can diagonalise e^{-itH} by conjugating with an appropriate matrix which is a tensor product of single-qubit unitaries U_i, each of which diagonalises the corresponding Pauli matrix s_i (for example, $HXH = Z$). Then we obtain

$$e^{-itH} = (U_1 \otimes U_2 \otimes \cdots \otimes U_n) e^{-i\alpha t z_1 \otimes z_2 \otimes \cdots \otimes z_n} (U_1^\dagger \otimes U_2^\dagger \otimes \cdots \otimes U_n^\dagger)$$

where $z_i \in \{I, Z\}$. As $e^{M \otimes I} = e^M \otimes I$ for any matrix M (which again follows from the definition), it remains to implement $e^{-i\alpha t Z \otimes Z \otimes \cdots \otimes Z}$ for some string of Z matrices of length $k \leq n$. For a computational basis state $|x\rangle$, where $x \in \{0,1\}^k$, this operation performs the map

$$|x\rangle \mapsto \begin{cases} e^{-i\alpha t}|x\rangle & \text{if } \sum_i x_i \text{ is even} \\ e^{i\alpha t}|x\rangle & \text{if } \sum_i x_i \text{ is odd} \end{cases}.$$

This can be achieved via the following quantum circuit of 2-qubit gates, which we illustrate for $k = 4$, but it should be clear how to generalise:



In this diagram,

$$R_z(\theta) = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix};$$

this is closely related to the R_d gate discussed in the context of the quantum Fourier transform. To see why this works, observe that the first sequence of CNOT gates sets the last qubit to $x_1 \oplus x_2 \oplus \cdots \oplus x_k$, the R_z gate performs the required phase shift depending on this quantity, and the remaining CNOT gates reset the qubits to their initial state (aside from this phase shift).

So we see that $O(n)$ quantum gates are sufficient to exactly simulate time-evolution according to an arbitrary Pauli matrix. This immediately generalises to the case where H is a weighted sum of m Pauli matrices that all commute (e.g. $H = X \otimes X + Z \otimes Z$), as in this special case

$$e^{-iHt} = e^{-it\sum_{j=1}^m \alpha_j \sigma_{s_j}} = \prod_{j=1}^m e^{-it\alpha_j \sigma_{s_j}},$$

so we can combine each of the individual simulations to simulate time-evolution according to H using $O(mn)$ quantum gates.

7.2 The non-commuting case

Unfortunately, this simulation technique does *not* necessarily work for non-commuting H_j. The reason is that if A and B are non-commuting operators, it need not hold that $e^{-i(A+B)t} = e^{-iAt}e^{-iBt}$. However, we can simulate non-commuting Hamiltonians via an observation known as the Lie-Trotter product formula. The technique will ultimately be based on combining approximate simulations of the operator $e^{-iHt/p}$ for some large integer p. We will first need a technical lemma describing how errors behave when we concatenate approximate simulations.

Lemma 7.1. *Let (U_i), (V_i) be sequences of m unitary operators satisfying $\|U_i - V_i\| \leq \epsilon$ for all $1 \leq i \leq m$. Then $\|U_m \dots U_1 - V_m \dots V_1\| \leq m\epsilon$.*

Proof. The proof is by induction on m. The claim trivially holds for $m = 1$. Assuming that it holds for a given m, we have

$$\begin{aligned}
&\|U_{m+1}U_m \dots U_1 - V_{m+1}V_m \dots V_1\| \\
&= \|U_{m+1}U_m \dots U_1 - U_{m+1}V_m \dots V_1 + U_{m+1}V_m \dots V_1 - V_{m+1}V_m \dots V_1\| \\
&\leq \|U_{m+1}U_m \dots U_1 - U_{m+1}V_m \dots V_1\| + \|U_{m+1}V_m \dots V_1 - V_{m+1}V_m \dots V_1\| \\
&= \|U_{m+1}(U_m \dots U_1 - V_m \dots V_1)\| + \|(U_{m+1} - V_{m+1})V_m \dots V_1\| \\
&= \|U_m \dots U_1 - V_m \dots V_1\| + \|U_{m+1} - V_{m+1}\| \\
&\leq (m+1)\epsilon.
\end{aligned}$$

□

Thus, in order to approximate $\prod_{j=1}^m e^{-iH_j t}$ to within ϵ, it suffices to approximate $e^{-iH_j t}$ for each j to within ϵ/m. Next we show how approximating a product of the form $\prod_j e^{-iH_j t}$ allows a sum of the form $e^{-i(\sum_j H_j)t}$ to be approximated.

In what follows, the notation $A + O(\epsilon)$, for a matrix A, is used as shorthand for $A + E$, where E is a matrix satisfying $\|E\| \leq C\epsilon$, for some universal constant C (not depending on A or ϵ).

Lemma 7.2 (Lie-Trotter product formula)**.** *Let A and B be Hermitian matrices such that $\|A\| \leq \delta$ and $\|B\| \leq \delta$, for some real $\delta \leq 1$. Then*

$$e^{-iA}e^{-iB} = e^{-i(A+B)} + O(\delta^2).$$

Proof. From the Taylor series for e^x, for any matrix A such that $\|A\| = \delta \leq 1$, we have

$$e^{-iA} = I - iA + \sum_{k=2}^{\infty} \frac{(-iA)^k}{k!} = I - iA + (-iA)^2 \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} = I - iA + O(\delta^2),$$

where the last equality follows from

$$\left\| \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} \right\| \leq \sum_{k=0}^{\infty} \frac{\delta^k}{(k+2)!} \leq e^{\delta} = O(1).$$

Hence

$$e^{-iA}e^{-iB} = \left(I - iA + O(\delta^2)\right)\left(I - iB + O(\delta^2)\right) = I - iA - iB + O(\delta^2) = e^{-i(A+B)} + O(\delta^2).$$

□

Applying this formula multiple times, for any Hermitian matrices $H_1, \dots, H_m$ satisfying $\|H_j\| \leq \delta \leq 1$ for all j,

$$\begin{aligned}
e^{-iH_1}e^{-iH_2} \dots e^{-iH_m} &= \left(e^{-i(H_1+H_2)} + O(\delta^2)\right) e^{-iH_3} \dots e^{-iH_m} \\
&= \left(e^{-i(H_1+H_2+H_3)} + O((2\delta)^2)\right) e^{-iH_4} \dots e^{-iH_m} + O(\delta^2) \\
&= e^{-i(H_1+\dots+H_m)} + O(\delta^2) + O((2\delta)^2) + \dots + O(((m-1)\delta)^2) \\
&= e^{-i(H_1+\dots+H_m)} + O(m^3\delta^2).
\end{aligned}$$

Write $H = \sum_{j=1}^{m} H_j$, where $\|H_j\| \leq \Delta$ (for example, H_j could be a weighted Pauli matrix as above; then this constraint corresponds to $|\alpha_j| \leq \Delta$). Applying this claim to the matrices $H_j t/p$ for arbitrary t and some large integer $p \geq t\Delta$, we have

$$\left\| e^{-iH_1 t/p} e^{-iH_2 t/p} \dots e^{-iH_m t/p} - e^{-i(H_1 + \dots + H_m)t/p} \right\| = O\left(m^3 \left(\frac{t\Delta}{p} \right)^2 \right).$$

So there is a universal constant C such that if $p \geq Cm^3 (t\Delta)^2/\epsilon$,

$$\left\| e^{-iH_1 t/p} e^{-iH_2 t/p} \dots e^{-iH_m t/p} - e^{-i(H_1 + \dots + H_m)t/p} \right\| \leq \epsilon/p.$$

By Lemma 7.1, for any such p,

$$\left\| \left(e^{-iH_1 t/p} e^{-iH_2 t/p} \dots e^{-iH_m t/p} \right)^p - e^{-i(H_1 + \dots + H_m)t} \right\| \leq \epsilon.$$

Given this result, we can simulate a Hamiltonian for time t simply by simulating the evolution of each of its terms for time t/p and concatenating the individual simulations; and if each term is a weighted Pauli matrix, we can use the results of the previous section for this. We summarise this as the following theorem.

Theorem 7.3. *Let H be a Hamiltonian which can be written as $H = \sum_{s \in \{I,X,Y,Z\}^n} \alpha_s \sigma_s$, where at most m coefficients α_s are nonzero, and $\max_s |\alpha_s| = O(1)$. Then, for any t, there exists a quantum circuit which approximates the operator e^{-iHt} to within ϵ in time $O(m^4 n t^2/\epsilon)$.*

It seems somewhat undesirable that, in order to simulate a Hamiltonian for time t, this algorithm has dependence on t which is $O(t^2)$. In fact, using more complicated simulation techniques, the dependence on t in Theorem 7.3 can be improved to linear, up to logarithmic factors.