MPRI 2.34.2: Quantum Information and Cryptography **2024 - 2025**

Lecture 4: Hamiltonian simulation

Lecturer: Simon Apers (apers@irif.fr)

Last lecture highlighted the importance of ground states of given Hamiltonians, as well as the utility of "integrating" Schrödinger's equation, i.e., implementing the evolution

$$|\psi\rangle \longmapsto e^{iHt}|\psi\rangle$$

for some initial state $|\psi\rangle$, time t and Hamiltonian H. While this evolution is native in "analog" quantum computers, in this lecture we will see how to implement this evolution in the more common quantum circuit model. This task is referred to as "Hamiltonian simulation", and its study has been a driving force in the field of quantum algorithms.

Conceptually, Hamiltonian simulation builds on the following observations:

1. Many Hamiltonians H can be split up into "simpler" building blocks $H = \sum_j H_j$.
2. There is a way of combining Hamiltonian simulation of the building blocks e^{iH_jt} to obtain the evolution e^{iHt}.

1 Pauli decomposition

One useful way of breaking up a Hamiltonian into simpler terms is by using the Pauli basis. The following exercise shows that we can split up any Hamiltonian into a sum of Pauli strings.

Exercise 1 (Pauli basis)**.** *Recall the unitary Pauli matrices*

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

- *Show that the Pauli basis $\{I, X, Y, Z\}$ forms a basis for the complex 2-by-2 matrices. I.e., any $A \in \mathbb{C}^{2\times 2}$ can be expanded as $A = \alpha_1 I + \alpha_x X + \alpha_y Y + \alpha_z Z$.*
- *Argue that this implies that the n-qubit Pauli basis $\{I, X, Y, Z\}^{\otimes n}$ forms a basis for the 2^n-by-2^n matrices.*

Such decomposition will be most useful when the number of Pauli strings is small, say $\text{poly}(n)$. This naturally occurs when the interactions are "local", as is demonstrated in the following exercise. The exercise is motivated by the use of the quantum adiabatic algorithm for the canonical "max cut" problem from combinatorial optimization, which requires efficient simulation of the max cut Hamiltonian.

Exercise 2 (Max cut)**.** *For a graph G with vertex set $[n]$ and (symmetric) edge set $E \subseteq [n]^2$, a maximum cut is described by a subset $Z \subset [n]$ that cuts a maximum number of edges (edges crossing from Z to Z^c). Equivalently, it maximizes the cut function*

$$c(A) = \sum_{(i,j)\in E} I_{i\in A, j\notin A}.$$

- *Identify a subset A with the indicator $a \in \{0,1\}^n$ ($i \in A \Leftrightarrow a_i = 1$). Express the cut function $c(A)$ as a degree-2 polynomial h in a.*
- *Rewrite the max cut Hamiltonian $H_1 = -\sum_{z\in\{0,1\}^n} c(z)\,|z\rangle\langle z|$ in terms of identity and Pauli-Z matrices.*

This exercise implies that the Hamiltonian showing up in the quantum adiabatic algorithm for max cut, $H(s) = (1-s)H_0 + sH_1$, can be decomposed into $O(n^2)$ Pauli strings, each of which acts nontrivial on at most 2 qubits.

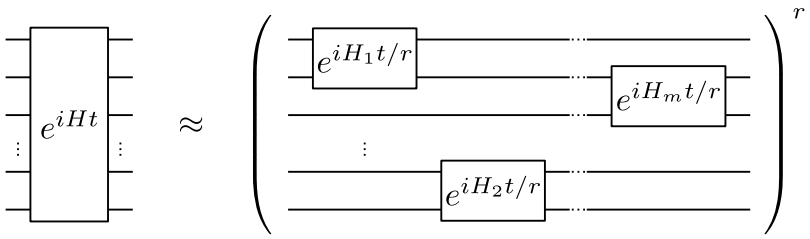
2 Hamiltonian simulation

From the previous section, we know that any Hamiltonian H can be expanded as a sum of simpler Hamiltonians, $H = \sum_{j=1}^m H_j$. E.g., for the max cut Hamiltonian (or in fact any "2-local" Hamiltonian), H consists of $m \in O(n^2)$ 2-qubit terms. Consequently, for any such H_j, Hamiltonian evolution $e^{iH_j t}$ is a simple 2-qubit gate, and we will assume access to these gates. The key question, then, is how do we combine the building blocks $\{e^{iH_j t}\}_{j=1}^m$ to obtain e^{iHt}?

Unfortunately, if the H_j's do not commute, then we do not have that $e^{iHt} = e^{iH_1 t}\dots e^{iH_m t}$. However, by the so-called "Lie-Trotter-Suzuki product formula", we do have that

$$e^{iHt} = \lim_{r\to\infty}\left(e^{iH_1 t/r}e^{iH_2 t/r}\dots e^{iH_m t/r}\right)^r.$$

For finite r, this corresponds to a quantum circuit with mr gates of the form $e^{iH_j t/r}$:



In the following exercise we bound the error incurred for finite r, focusing on the special case where $m = 2$.

Exercise 3 (Trotterization)**.**

- *Assume $\|A_1\|, \|A_2\| \le 1$. Use the Taylor series $e^B = I + B + O(\|B\|^2)$ for $\|B\| \le 1$ to show that*

$$e^{A_1+A_2} = e^{A_1}e^{A_2} + E, \quad \|E\| \in O(\|A_1\|^2 + \|A_2\|^2). \tag{1}$$

- *Assume $\|H_1\|, \|H_2\| \le 1$. For $r \ge t$, argue that*

$$e^{i(H_1+H_2)t} = \left(e^{iH_1 t/r}e^{iH_2 t/r}\right)^r + E_r, \quad \|E_r\| \in O(t^2/r).$$

Picking $r \in O(t^2/\epsilon)$, it follows that we can ϵ-approximate the Hamiltonian evolution e^{iHt} using $r \in O(t^2/\epsilon)$ 2-qubit gates. More generally, when $m > 2$, the scaling becomes $O(m^3t^2/\epsilon)$, which remains efficient as long as m is poly(n).

We conclude by tying this back to the previous lecture, in which we saw how a circuit model computation can be encoded into an adiabatic computation. Hamiltonian simulation makes this exercise in the opposite direction: it shows that an adiabatic computation (more precisely, evolution by a Hamiltonian) can be simulated in the circuit model. In a particular sense, this shows that both computational models are equivalent.