CS 599 P1: Introduction to Quantum Computation Boston University, Fall 2025
Instructor: Alexander Poremba

LECTURE # 17: QUANTUM SIMULATION

1 Quantum Simulation

In this lecture, we encounter arguable one of the most important applications of quantum computers; namely, *Hamiltonian simulation*. This problem concerns how to efficiently approximate the continuous time evolution of a quantum system using a discrete quantum circuit. While the Schrödinger equation describes dynamics in principle, implementing these dynamics on a digital quantum computer requires nontrivial approximations, analysis of errors, and careful use of the structure of the Hamiltonian.

Quantum simulation lies at the intersection of physics, linear algebra, numerical analysis, and quantum computation. In physics, the evolution operator e^{-iHt} predicts how a system behaves over time. In quantum algorithms, it becomes a primitive for higher-level procedures such as quantum phase estimation, ground-state energy estimation, and quantum chemistry algorithms. Understanding how to simulate e^{-iHt} is therefore essential for both foundational and applied quantum computing.

We begin by reviewing the mathematical structure of Hermitian matrices and matrix exponentials, before turning to the simulation problem itself.

2 Background: Hermitian matrices and spectral decomposition

A *Hamiltonian* in quantum mechanics is simply a Hermitian matrix. Physically, H describes the total energy of a closed quantum system, and the spectral properties of H dictate both the measurable energy levels of the system and how it evolves in time.

The appeal of Hermitian matrices comes from the spectral theorem, which provides a robust structure theorem and turns many analytic questions into simple manipulations of eigenvalues.

Definition 2.1 (Hermitian matrix)**.** *A matrix $H \in \mathbb{C}^{N\times N}$ is* Hermitian *if*

$$H^\dagger = H.$$

Hermitian matrices always have real eigenvalues and an orthonormal basis of eigenvectors. This is crucial in quantum mechanics: the eigenvalues correspond to physical energy levels, and the orthonormal eigenvectors form a complete basis of energy eigenstates.

Spectral theorem for Hermitian matrices

If $H = H^\dagger$, then there exist real numbers $E_0, \dots, E_{N-1}$ and orthonormal vectors $\{|v_j\rangle\}$ such that

$$H = \sum_{j=0}^{N-1} E_j \,|v_j\rangle \langle v_j| \,.$$

Matrix functions and exponentials

Many functions of real variables extend naturally to Hermitian matrices. If H admits the above spectral decomposition, then for any function $f : \mathbb{R} \to \mathbb{C}$,

$$f(H) := \sum_{j=0}^{N-1} f(E_j) \left|v_j\right\rangle\!\left\langle v_j\right| .$$

The matrix exponential is the most important example:

$$e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} = \sum_j e^{-iE_j t} \left|v_j\right\rangle\!\left\langle v_j\right| .$$

This expression makes it clear that the dynamics generated by H are easy to understand in the eigenbasis of H: each energy eigenstate simply accumulates a phase at a rate proportional to its eigenvalue.

3 Quantum Simulation

Quantum simulation asks whether such natural quantum dynamics can be reproduced by a quantum computer in an efficient way.

Motivation

Simulating quantum systems is one of the oldest motivations for quantum computing, going back to Feynman's 1982 proposal. Many-body systems exhibit an exponentially large state space, making classical simulation intractable. Quantum simulation aims to circumvent this barrier.

A quantum computer stores and manipulates quantum states directly, so perhaps it can efficiently emulate any physically reasonable quantum system.

The following diagram illustrates the goal:

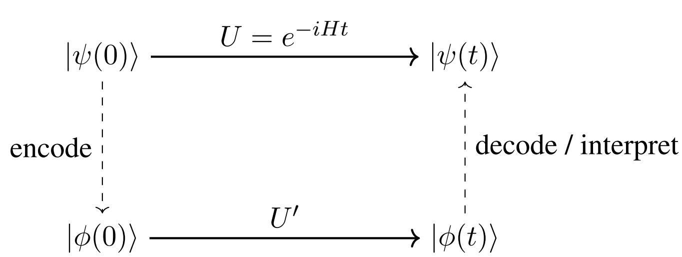

Figure 1: Quantum simulation as an approximate commutative diagram.

A successful simulation produces a state $|\phi(t)\rangle$ that encodes the same physical information as $|\psi(t)\rangle$ up to allowable precision.

3.1 The Schrödinger equation and its solution

In nonrelativistic quantum mechanics, the evolution of a closed quantum system obeys the Schrödinger equation (with $\hbar = 1$):

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

Because this is a first-order linear differential equation, its solution is given by the matrix exponential:

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

Using the spectral decomposition of H,

$$e^{-iHt} = \sum_j e^{-iE_j t}|v_j\rangle\langle v_j| .$$

Proposition: e^{-iHt} is unitary

If H is Hermitian, then $U(t) = e^{-iHt}$ satisfies $U(t)^\dagger U(t) = I$.

This ensures that time evolution preserves inner products and probabilities, as required by the postulates of quantum mechanics.

3.2 The Hamiltonian simulation problem

Formally, the goal of Hamiltonian simulation is to approximate this physical time evolution using a finite quantum circuit.

Problem (Hamiltonian simulation)

- **Input:** A decomposition $H = H_1 + \cdots + H_m$, time t, precision $\varepsilon > 0$, and an initial state $|\psi(0)\rangle$.
- **Output:** A state $|\psi'\rangle$ such that

$$\|\,|\psi'\rangle - e^{-iHt}|\psi(0)\rangle\,\| < \varepsilon.$$

Such procedures form the backbone of a large class of quantum algorithms in chemistry and condensed matter physics.

4 Tools for Hamiltonian Simulation

4.1 Warm-up: A single k-local term

Suppose the Hamiltonian consists of a single term acting on only k of the n qubits:

$$H_1 = h \otimes I_{n-k}.$$

Then

$$e^{-iH_1 t} = e^{-iht} \otimes I_{n-k}.$$

Since e^{-iht} is a k-qubit unitary, it can be decomposed into $2^{O(k)}$ one- and two-qubit gates. Thus if $k = O(1)$, such evolutions are easy to implement.

Most physically relevant Hamiltonians are sums of such local terms, which motivates the next question: can we factor the global evolution into a product of local evolutions?

4.2 Why naive factorization fails

Given $H = H_1 + H_2$, it is tempting to write

$$e^{-iHt} = e^{-iH_1 t} e^{-iH_2 t}.$$

This equality holds for real numbers, but matrices do not necessarily commute.

The Pauli matrices X and Z provide a simple counterexample:

$$XZ \neq ZX.$$

Definition 4.1 (Commutator)**.** *For square matrices A, B, the commutator is*

$$[A, B] := AB - BA.$$

Fact

$e^{A+B} = e^A e^B$ if and only if $[A, B] = 0$.

Most realistic Hamiltonians contain many noncommuting terms, so exact factorization is typically impossible. Nevertheless, approximate factorizations *are* possible and form the basis of Trotter–Suzuki methods.

5 The Trotter–Suzuki Approach

5.1 The Trotter formula

Lie–Trotter product formula

For any matrices A and B,

$$e^{A+B} = \lim_{r \to \infty} \left(e^{A/r} e^{B/r} \right)^r.$$

Applied to Hamiltonians:

$$e^{-i(H_1+H_2)t} = \lim_{r \to \infty} \left(e^{-iH_1 t/r} e^{-iH_2 t/r} \right)^r.$$

5.2 Many-term version

Approximation bound for many terms

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

$$\left\| e^{-iHt} - \left(e^{-iH_1 t/r} \cdots e^{-iH_m t/r} \right)^r \right\| \le \varepsilon,$$

whenever

$$r \gg \frac{m^2 t^2 (\max_j \|H_j\|^2)}{\varepsilon}.$$

The operator norm here is

$$\|A\| := \max_{\|\psi\|=1} \|A \left|\psi\right\rangle \|.$$

6 Hamiltonian Simulation Algorithm

We can now describe the algorithm for simulating e^{-iHt} using Trotterization.

Hamiltonian simulation algorithm

- **Input:** A k-local Hamiltonian $H = H_1 + \cdots + H_m$, time t, precision ε, and initial state $|\psi(0)\rangle$.
- **Choose:**

$$r \gg \frac{m^2 t^2 (\max_i \|H_i\|^2)}{\varepsilon}.$$

- **For** r **repetitions:** Apply the product

$$e^{-iH_1 t/r},\ e^{-iH_2 t/r},\ \ldots,\ e^{-iH_m t/r}.$$

- **Output:** A state $|\psi'\rangle$ satisfying

$$\| \left|\psi'\right\rangle - e^{-iHt} \left|\psi(0)\right\rangle \| < \varepsilon.$$

Complexity. Each $e^{-iH_j t/r}$ is k-local and can be implemented with $2^{O(k)}$ gates. Thus the overall complexity is

$$\text{Gate count} = 2^{O(k)} \cdot m \cdot r = \text{poly}\left(2^k, m, t, \max_i \|H_i\|, 1/\varepsilon\right).$$

Remark (Beyond Trotterization). While Trotter–Suzuki methods are intuitive and widely used, the last decade has seen major algorithmic advances. Techniques such as *qubitization*, *quantum signal processing*, and *linear combinations of unitaries* achieve asymptotically optimal complexity. Despite this, Trotter methods remain a valuable pedagogical starting point and continue to be competitive for near-term implementations.