Hamiltonian Simulation

Quantum Computing

Álvaro Sánchez-Paniagua, Ana Vargas, Gonzalo Martínez de Sola, Iñigo Iriondo, Miguel Laredo

Based on the notes from Ashley Montaro

April 3, 2025

Motivation

Introduction: Hamiltonian Simulation

Hamiltonian

Hermitian operator acting on n qubits which corresponds physically to a system made up of n 2-level subsystems.

Schrödinger's equation

Time evolution of the state $|\psi\rangle$ of a quantum system:

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

Introduction: Hamiltonian Simulation

Solution of the Schrödinger's equation (*time-independent*):

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

Goal

To approximate the unitary operator:

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

Approximation in the operator (spectral) norm:

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

$\tilde{U}$ approximates U within ϵ if

$$||\tilde{U} - U|| < \epsilon$$

Quantum Simulation of a Hamiltonian

Objective: Efficiently simulate a Hamiltonian H using a quantum circuit. Specifically, approximate the time evolution operator:

$$U = e^{-iHt}$$

with a quantum circuit composed of a polynomial number of gates (poly(n)).

Focus: We consider Hamiltonians acting on a Hilbert space of dimension 2^n, i.e., matrices of size $2^n \times 2^n$.

Key Question: How can a Hamiltonian, which has exponentially many parameters, be approximated by a quantum circuit that only contains polynomially many parameters (poly(n) gates)?

Hamiltonians as Sums of Pauli Matrices

Motivation: To efficiently simulate quantum systems, we leverage certain physical properties that allow for an efficient decomposition.

Pauli Decomposition of Hamiltonians: We consider Hamiltonians expressed as sums of Pauli matrices acting on n qubits.

Notation: For $s \in \{I, X, Y, Z\}^n$, we define the matrix σ_s as the tensor product of the corresponding Pauli matrices:

$$\sigma_s = s_1 \otimes s_2 \otimes \cdots \otimes s_n.$$

For example:

$$\sigma_{ZX} = Z \otimes X.$$

Representation of Hamiltonians

Key Idea: Any Hermitian $2^n \times 2^n$ matrix can be decomposed as a sum of Pauli matrices like we previously said:

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

where the coefficients α_s are real numbers.

Why is this useful and makes any sense?

- Pauli matrices form a basis for all $2^n \times 2^n$ Hermitian matrices.
- If only a polynomial number of coefficients are nonzero, then H can be simulated efficiently.
- This sparsity is not arbitrary—it corresponds to physical Hamiltonians with local interactions.

Step 1: The Pauli Matrices as a Basis

Pauli Matrices (for 1 qubit):

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

Generalized Pauli Basis (for n qubits):

$$\sigma_s = s_1 \otimes s_2 \otimes \cdots \otimes s_n, \quad s_i \in \{I, X, Y, Z\}$$

Goal: Prove these matrices form an orthonormal basis for $2^n \times 2^n$ Hermitian matrices.

Key Steps:

- Show linear independence.
- Prove they span the entire space.
- Use the Hilbert-Schmidt inner product:

$$\langle A, B \rangle = \frac{1}{2^n} \mathrm{Tr}(A^\dagger B)$$

Why the Trace?

Intuition: The Trace as a Similarity Check

Analogy: The trace $\mathrm{Tr}(A^\dagger B)$ acts like a "dot product" for matrices.

Example (1 qubit):

$$\mathrm{Tr}(XZ) = \mathrm{Tr}\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\right) = \mathrm{Tr}\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} = 0 + 0 = 0.$$

$$\mathrm{Tr}(XX) = \mathrm{Tr}(I) = 2.$$

Key Insight:

- Different Pauli matrices: $\mathrm{Tr}(S_i S_j) = 0$ (orthogonal).
- Same Pauli matrices: $\mathrm{Tr}(S_i S_j) = 2$.

General Case: For $S_i \neq S_j$, $\mathrm{Tr}(S_i S_j) = 0 \implies$ No overlap $\implies$ Independent!

Step 2: Proving Orthogonality

Property: For single-qubit Pauli matrices S_i, S_j:

$$\mathrm{Tr}(S_i S_j) = 2\delta_{ij}, \quad \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

**Why δ_{ij}? ** It encodes orthogonality:

- If $S_i \neq S_j$, their product is traceless (e.g., $XZ = -iY$, $\mathrm{Tr}(-iY) = 0$).
- If $S_i = S_j$, $S_i^2 = I \implies \mathrm{Tr}(I) = 2$.

For Tensor Products: Orthogonality scales to n qubits:

$$\mathrm{Tr}(\sigma_s \sigma_t) = \prod_{k=1}^{n} \mathrm{Tr}(s_k t_k) = \begin{cases} 2^n & \text{if } \sigma_s = \sigma_t, \\ 0 & \text{otherwise.} \end{cases}$$

Step 3: Why Tensor Products Work

Key Idea: Orthogonality propagates through tensor products.

Example: Let $\sigma_{XZ} = X \otimes Z$, $\sigma_{IX} = I \otimes X$.

$$\mathrm{Tr}(\sigma_{XZ}\sigma_{IX}) = \mathrm{Tr}(X \otimes Z \cdot I \otimes X) = \mathrm{Tr}(X \cdot I) \otimes \mathrm{Tr}(Z \cdot X) = \mathrm{Tr}(X) \cdot \mathrm{Tr}(ZX) = 0 \cdot 0 = 0.$$

Result: All σ_s matrices are orthogonal!

Step 4: Why This Works for Hamiltonians

Completeness: There are 4^n Pauli tensor products, matching the dimension of $2^n \times 2^n$ matrices and also being orthogonal:

$$\dim(\text{Hermitian matrices}) = 4^n.$$

Expanding Hamiltonians: Therefore, any H can be written as a linear combination of our Pauli matrices:

$$H = \sum_s \alpha_s \sigma_s, \quad \alpha_s = \frac{1}{2^n} \mathrm{Tr}(H\sigma_s).$$

Conclusion: The Pauli basis spans the space $\implies$ Hamiltonians live here!

Summary: The Pauli Basis in a Nutshell

- **Orthogonality and linear independence:** $\mathrm{Tr}(\sigma_s \sigma_t) = 2^n \delta_{st}$.
- **Completeness:** 4^n matrices $\equiv$ dimension of the space.
- **Hamiltonians:** Decompose into weighted sums of Pauli matrices.

$$\boxed{\text{The Pauli tensor products form a natural basis for quantum operators!}}$$

Representation of Hamiltonians: k-local Hamiltonians

Local interactions in quantum systems are captured by a class of Hamiltonians known as **k-local Hamiltonians**. These are written as:

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

where each H_j is a Hermitian matrix that acts non-trivially on at most k qubits while acting as the identity on the remaining $n - k$ qubits.

Example: Consider a Hamiltonian on 3 qubits that is 2-local. One possible definition is:

$$H = \underbrace{X \otimes I \otimes I}_{\substack{\text{Acts on qubit 1} \\ \text{Identity on qubits 2,3}}} - \underbrace{2I \otimes Z \otimes Y}_{\substack{\text{Acts on qubits 2,3} \\ \text{Identity on qubit 1}}}.$$

This local structure (interactions affecting only a few qubits) is key to designing efficient quantum simulations, as it allows us to approximate the overall evolution using a circuit with only a polynomial number of gates.

The 2D Ising Model: Conceptual Introduction

What Is the 2D Ising Model?

The 2D Ising model is a fundamental model in statistical physics that describes how microscopic magnetic moments interact on a two-dimensional grid.

Key Concepts:

- **Spins:** These are the basic magnetic units (or qubits in a quantum context) that can be in one of two states—commonly referred to as "up" or "down". They represent the magnetic moment of an atom.
- **Local Interactions:** Each spin interacts only with its nearest neighbors. This local coupling captures how individual magnetic moments influence each other.
- **Global Behavior:** Despite the simplicity of local interactions, the model exhibits rich behavior such as collective magnetism (ferromagnetism) and phase transitions.

Expressing the Ising Hamiltonian with Pauli Operators

Full Expansion: For an $n \times n$ lattice:

$$H = J \sum_{i,j=1}^{n} \Big(\underbrace{Z(i,j) \otimes Z(i,j+1)}_{\text{horizontal interaction}} + \underbrace{Z(i,j) \otimes Z(i+1,j)}_{\text{vertical interaction}} \Big)$$

Tensor Product Example: $Z(1,1) \otimes Z(1,2) \otimes I \otimes \cdots \otimes I$ (only 2-local terms)

Physical Intuition:
Each term couples only a qubit and its nearest neighbor (both horizontally and vertically), reflecting the local interactions of the Ising model.

Proportional Case

Hamiltonian is proportional to Pauli Tensor Product

We consider the simple case where H is proportional to a Pauli matrix on n qubits:

$$H = \alpha s_1 \otimes s_2 \otimes \cdots \otimes s_n{}^{a}$$

[a] H is a 2 n × 2 n matrix, it contains exponentially many parameters

Time Evolution Operator (What we aim to compute)

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

Matrix Exponential

For any square matrix $A \in \mathbb{C}^{n \times n}$, the matrix exponential is defined by:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots$$

Diagonalizing H

Walkthrough of how to make H diagonal

$$H = \alpha s_1 \otimes s_2 \otimes \cdots \otimes s_n.$$

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

- **Identity**:
$$III^\dagger = I$$
- **Pauli-Z**:
$$IZI^\dagger = Z$$
- **Pauli-X**:
$$HXH^\dagger = Z$$
- **Pauli-Y**:
$$U_Y Y U_Y^\dagger = Z \quad \text{where} \quad U_Y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}$$

Diagonalization of H

$$
\begin{aligned}
H &= \alpha(s_1 \otimes \cdots \otimes s_n) \\
&= \alpha(U_1 z_1 U_1^\dagger) \otimes \cdots \otimes (U_n z_n U_n^\dagger) \quad \text{(Diagonalize each } s_i\text{)} \\
&= \alpha(U_1 \otimes \cdots \otimes U_n)(z_1 \otimes \cdots \otimes z_n)(U_1^\dagger \otimes \cdots \otimes U_n^\dagger)
\end{aligned}
$$

And thus,

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

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

The conjugation property of the matrix exponential

$$e^{UHU^\dagger} = Ue^H U^\dagger$$

This identity is easily proven using the power series definition for matrix exponential.

we diagonalize e^{-itH} with an appropriate unitary transformation:

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

where $z_i \in \{I, Z\}$, seen earlier in the slides.

Implementation via Quantum Circuits

Since $e^{M\otimes I} = e^M \otimes I$ The problem reduces to implementing the following

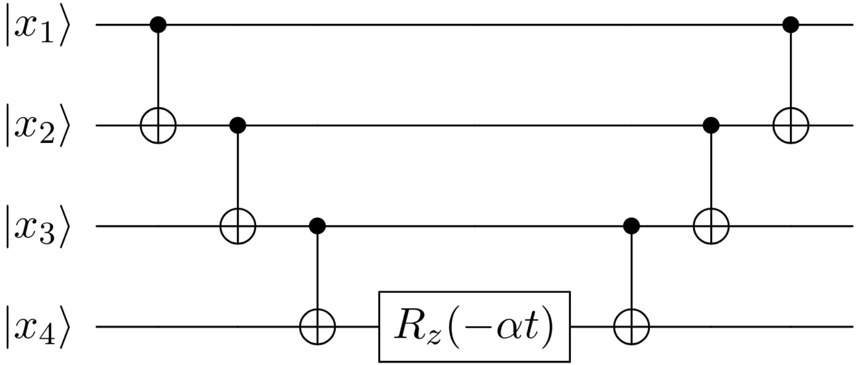
$$e^{-i\alpha t Z\otimes Z\otimes\cdots\otimes Z}.$$

$$(Z \otimes Z \otimes \cdots \otimes Z)\,|x_1 x_2 \cdots x_k\rangle = (-1)^{x_1}(-1)^{x_2}\cdots(-1)^{x_k}\,|x_1 x_2 \cdots x_k\rangle$$

The action of the k-qubit Z-interaction unitary on a computational basis state is given by:

$$e^{-i\alpha t Z\otimes\cdots\otimes Z}|x\rangle = \begin{cases} e^{-i\alpha t}|x\rangle & \text{if } \sum_{i=1}^{k} x_i \text{ is even} \\ e^{i\alpha t}|x\rangle & \text{if } \sum_{i=1}^{k} x_i \text{ is odd} \end{cases}$$

Quantum Circuit for $k = 4$



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

Generalization to Weighted Sums of Pauli Matrices

For H as a weighted sum of commuting Pauli matrices:

$$H = \sum_{j=1}^{m} \alpha_j \sigma_{s_j}$$

the evolution follows as:

$$e^{-iHt} = \prod_{j=1}^{m} e^{-i\alpha_j \sigma_{s_j} t}$$

This requires $O(mn)$ quantum gates.

Non-Commuting Hamiltonians

If A and B are non-commuting operators, we might have that

$$e^{-i(A+B)t} \neq e^{-iAt} \, e^{-iBt} \, .$$

In this case we can combine approximate simulations of $e^{-iHt/p}$ for some large p. We will need two lemmas:

- Error in approximate simulation concatenation.
- Lie-Trotter product formula.

Lemma 1

Let $(U_i)i = 1^m$ and $(V_i)i = 1^m$ be sequences of unitary operators satisfying

$$\|U_i - V_i\| \leq \epsilon \quad \text{for all } 1 \leq i \leq m.$$

Then,

$$\|U_m U_{m-1} \cdots U_1 - V_m V_{m-1} \cdots V_1\| \leq m\,\epsilon.$$

Proof by induction:

- Base case: For $m = 1$, the claim is trivial.
- Inductive step: Assume the claim holds for m. We need to show that $\|U_{m+1} U_m \cdots U_1 - V_{m+1} V_m \cdots V_1\| \leq (m+1)\epsilon$.

Inductive step

$$
\begin{aligned}
&\|U_{m+1}U_m\cdots U_1 - V_{m+1}V_m\cdots V_1\| = \\
&\|U_{m+1}U_m\cdots U_1 - U_{m+1}V_m\cdots V_1 \\
&+U_{m+1}V_m\cdots V_1 - V_{m+1}V_m\cdots V_1\|
\end{aligned}
$$

Inductive step

$$
\begin{aligned}
\|U_{m+1}U_m\cdots U_1 - V_{m+1}V_m\cdots V_1\| &= \\
\|U_{m+1}U_m\cdots U_1 \color{red}{- U_{m+1}V_m\cdots V_1} \\
\color{red}{+U_{m+1}V_m\cdots V_1} - V_{m+1}V_m\cdots V_1\| &\leq \\
\|U_{m+1}U_m\cdots U_1 \color{red}{- U_{m+1}V_m\cdots V_1}\| \\
+\|\color{red}{U_{m+1}V_m\cdots V_1} - V_{m+1}V_m\cdots V_1\|
\end{aligned}
$$

Inductive step

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

Inductive step

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

Inductive step

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

Simulation technique

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 this allows a sum of the form

$$e^{-i\left(\sum_j H_j\right)t}$$

to be approximated.

Lemma 2

Lie-Trotter product formula
Let A and B be Hermitian matrices satisfying

$$\|A\| \leq \delta, \quad \|B\| \leq \delta, \quad \text{with } \delta \leq 1.$$

Then,

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

Notation: $E = O(\epsilon)$ means that the matrix satisfies $\|E\| \leq C\,\epsilon$ for some universal constant C.

Proof of Lemma 2

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

Proof of Lemma 2

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

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

Proof of Lemma 2

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

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

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

Proof of Lemma 2

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

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

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

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

Proof of Lemma 2

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

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

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

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

$$I - iA + O(\delta^2)\cancel{O(e^{\delta}})$$

Proof of Lemma 2

$$
\begin{aligned}
e^{-iA} = I - iA + \sum_{k=2}^{\infty} \frac{(-iA)^k}{k!} &= \\
I - iA + (-iA)^2 \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} &= \\
I - iA + O(\|(-iA)^2\|) O(\| \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} \|) &= \\
I - iA + O(\delta^2) O(\sum_{k=0}^{\infty} \frac{\delta^k}{(k+2)!}) &= \\
I - iA + O(\delta^2) \cancel{O(e^{\delta}}) &
\end{aligned}
$$

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

Hamiltonian Simulation Approximation

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.

Hamiltonian Simulation Approximation

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.

We can now apply **Lemma 2** formula multiple times, for any Hermitian matrices $H_1, \ldots, H_m$ satisfying $\|H_j\| \leq \delta \leq 1$ for all j as follows,

Hamiltonian Simulation Approximation

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.

We can now apply **Lemma 2** formula multiple times, for any Hermitian matrices $H_1, \ldots, H_m$ satisfying $\|H_j\| \leq \delta \leq 1$ for all j as follows,

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

Hamiltonian Simulation Approximation

Write $H = \sum_{j=1}^{m} H_j$, where $\|H_j\| \leq \Delta$. Applying this claim to matrices $H_j t/p$ for any t and $p \geq t\Delta$, we have:

Hamiltonian Simulation Approximation

Write $H = \sum_{j=1}^{m} H_j$, where $\|H_j\| \leq \Delta$. Applying this claim to matrices $H_j t/p$ for any t and $p \geq t\Delta$, we have:

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

Hamiltonian Simulation Approximation

Write $H = \sum_{j=1}^{m} H_j$, where $\|H_j\| \leq \Delta$. Applying this claim to matrices $H_j t/p$ for any t and $p \geq t\Delta$, we have:

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

As an example, H_j could be a weighted Pauli matrix as in the previous section, where the constraint corresponds to $|\alpha_j| \leq \Delta$.

Universal Constant and Approximation Bound

There exists a universal constant C such that for $p \geq Cm^3(t\Delta)^2/\epsilon$:

Universal Constant and Approximation Bound

There exists a universal constant C such that for $p \geq Cm^3(t\Delta)^2/\epsilon$:

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

Universal Constant and Approximation Bound

There exists a universal constant C such that for $p \geq Cm^3(t\Delta)^2/\epsilon$:

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

By **Lemma 1**, for any such p,

Universal Constant and Approximation Bound

There exists a universal constant C such that for $p \geq Cm^3(t\Delta)^2/\epsilon$:

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

By **Lemma 1**, for any such p,

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

Universal Constant and Approximation Bound

There exists a universal constant C such that for $p \geq Cm^3(t\Delta)^2/\epsilon$:

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

By **Lemma 1**, for any such p,

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

By this result, we can simulate a Hamiltonian for time t by simulating the evolution of each of this terms for time t/p and concatenating each of the simulations.

Hamiltonian Simulation Theorem

Theorem

Let H be a Hamiltonian 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 that approximates e^{-iHt} within ϵ in time:

$$O(m^4 n t^2 / \epsilon).$$

Computational Complexity Considerations

It seems undesirable that simulating a Hamiltonian for time t depends on t as $O(t^2)$. However, using more advanced techniques, this dependence can be improved to linear, up to logarithmic factors.

Recap

Key points we have followed in order to showcase Hamiltonian simulation:

- Defined a notion of simulation
- Defined the circuit to simulate a Hamiltonian proportional to a Pauli matrix. $O(n)$
- Extend this result to the case in which the Hamiltonian is a sum of m weighted Pauli matrices that commute. $O(nm)$
- Finally, see how to approximate the non commuting case.