7. Quantum algorithms

(i) Quantum algorithm for simulating quantum physics

Time evolution of a closed quantum system

Recall the postulate of QM saying that a quantum system evolves unitarily, or as experimentally observed, according to Schroedinger's equation:

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

where

$\hbar$ = Planck's constant (absorbed in H(t))

$|\psi(t)\rangle$ = state of the N-dim system at time t

H(t) = Hamiltonian of the system at time t.

hermitian NxN matrix, in units of energy

Hamiltonian simulation

Problem: Hamiltonian simulation

Given:
Initial state: $|\psi(0)\rangle$
Hamiltonian: $H(t)$ for $0 \leq t \leq T$
Time T

Output: a copy of the state $|\psi(T)\rangle$.

Hamiltonian simulation

Problem: Hamiltonian simulation

Given:
Initial state: $|\psi(0)\rangle$
Hamiltonian: $H(t)$ for $0 \le t \le T$
Time T

Output: a copy of the state $|\psi(T)\rangle$.

Variation: for a given measurement, obtain a sample of the outcome of applying the measurement to $|\psi(T)\rangle$ or collect statistics of the outcomes.

Motivation:

1. Understand physics

2. Understand electronic structures in large molecules and solid state systems

Potential applications:

Quantum field theory, quantum chromodynamics, condensed matter / many body physics.

Quantum chemistry (drugs, protein folding / diseases, photosynthesis) ??

Small quantum devices (lasers, quantum dots, mesoscopic physics, fabrications of materials).

Simulation of quantum dynamics is challenging:

For an N-dim system, Schroedinger's equation is a coupled system of N differential equations in t, and N is exponential in the number of "qubits" or constituent systems, like # atoms in a protein.

Simulation of quantum dynamics is challenging:

For an N-dim system, Schroedinger's equation is a coupled system of N differential equations in t, and N is exponential in the number of "qubits" or constituent systems, like # atoms in a protein.

Feynman raised the possibility in 1985 (clarified and made precise by others):

Can we efficiently simulate quantum dynamics (no choice over the given H(t) and initial state) using a quantum computer (say, a quantum circuit with standard initial states and manipulated by a universal set of gates?)

Caution!

Solving Schroedinger's equation gives classical description of the final state to a certain precision (which is exponential).

Hamiltonian simulation by a quantum computer provides only COPIES of the final quantum state or measurement outcomes. Don't ask for a description of the final state to avoid exponential complexity!

Caution!

Solving Schroedinger's equation gives classical description of the final state to a certain precision (which is exponential).

Hamiltonian simulation by a quantum computer provides only COPIES of the final quantum state or measurement outcomes. Don't ask for a description of the final state to avoid exponential complexity!

We focus on problems where we want to calculate something specific from the final state that can be done by measuring the final state, which motivates how we formulate the problem earlier.

Regime of interest:

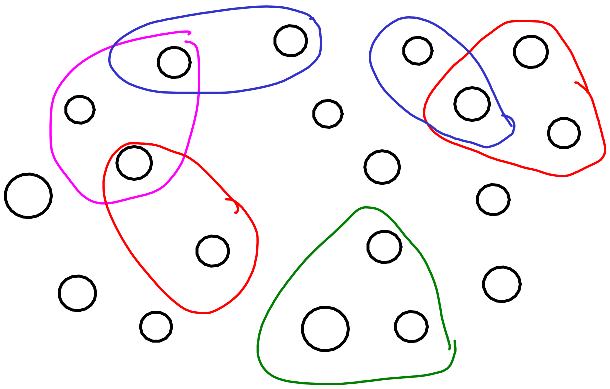
1. Hamiltonian is time independent.

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

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

Regime of interest:

2. Hamiltonian is "local", $H = \sum_{k=1}^{L} H_k$ where each H_k



only acts on a few systems

(eg $H_k = I \otimes I \otimes \cdots \otimes A \otimes I \otimes B \otimes I \otimes \cdots \otimes I$)

and $L \leq \text{poly}(n)$.

n systems, of dim $d_1, d_2, \ldots, d_n$, $N = d_1 d_2 \cdots d_n$.

Example: The Ising model

Here, we have n "spins" (qubits), each "interacting" only with nearest neighbors.

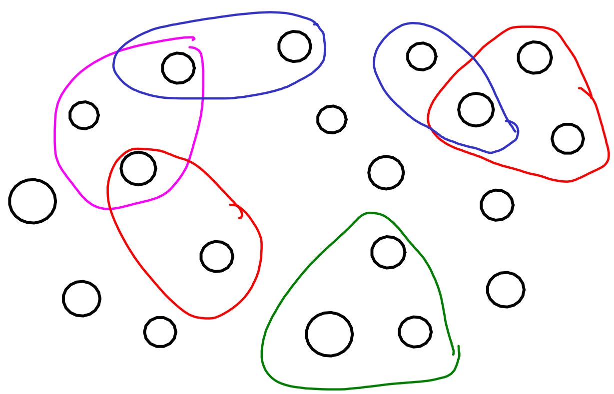
$$H = \sum_{k=1}^{n-1} \underbrace{X_k \otimes X_{k+1} + Y_k \otimes Y_{k+1} + Z_k \otimes Z_{k+1}}_{H_k} \qquad (\otimes I\text{'s omitted})$$

e^{-iHt} is hard to evaluate since $[H_k, H_{k+1}] \neq 0$

Here, [A,B] = AB - BA is the "commutator" of A, B.

Regime of interest:

2. Hamiltonian is "local", $H = \sum_{k=1}^{L} H_k$ where each H_k



only acts on a few systems

$(\text{eg } H_k = I \otimes I \otimes \cdots \otimes A \otimes I \otimes B \otimes I \otimes \cdots \otimes I)$

and $L \leq \text{poly}(n)$.

n systems, of dim $d_1, d_2, \ldots, d_n$, $N = d_1 d_2 \cdots d_n$.

Surprisingly, almost all quantum systems of interest have local Hamiltonians. In fact, most interactions in nature are 2-body interactions.

Main ideas behind an efficient quantum simulation

1. If the H_k's commute, then, $e^{-i \sum_k H_k t} = \prod_k e^{-i H_k t}$.

Main ideas behind an efficient quantum simulation

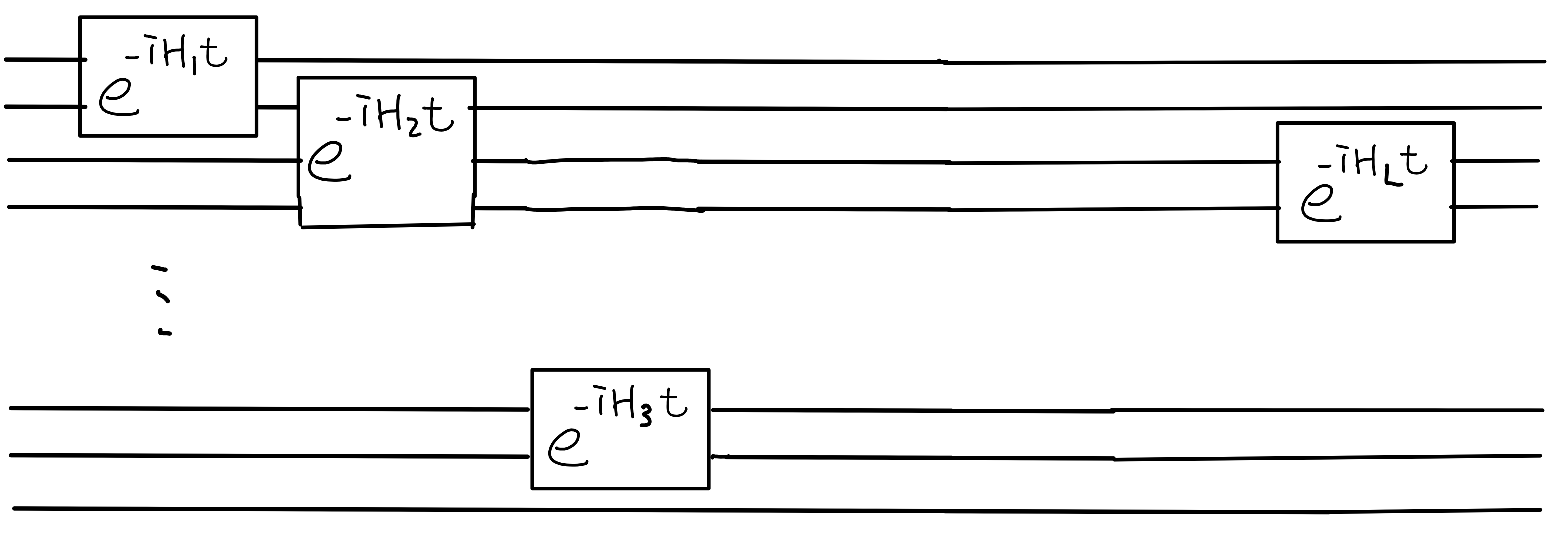
1. If the H_k's commute, then, $e^{-i\sum_k H_k t} = \prod_k e^{-iH_k t}$.

   i.e., the evolution due to $\sum_k H_k$ is composed of

   $e^{-iH_1 t} \; e^{-iH_2 t} \; \ldots$

   so we can perform each $e^{-iH_k t}$ separately.

e.g.



Main ideas behind an efficient quantum simulation

1. If the H_K's commute, then, $e^{-i \sum_K H_K t} = \prod_K e^{-iH_K t}$.

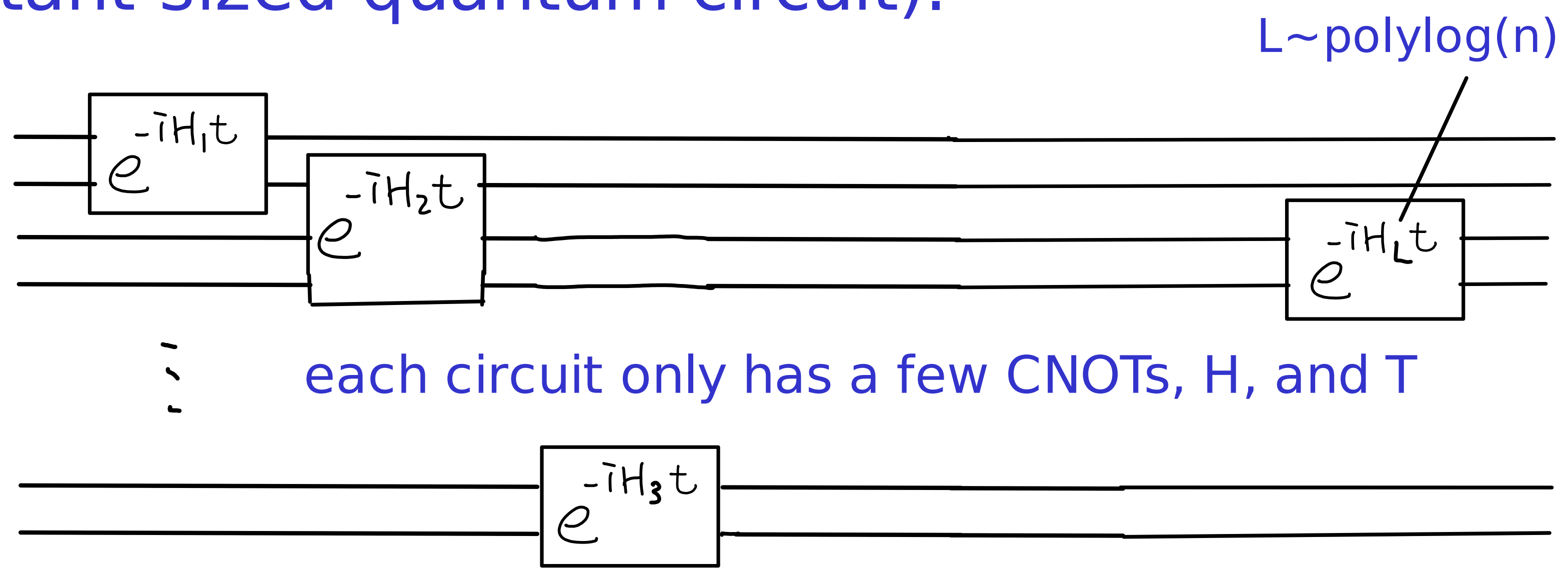
   i.e., the evolution due to $\sum_K H_K$ is composed of

   $e^{-iH_1 t} \; e^{-iH_2 t} \; \ldots$

   so we can perform each $e^{-iH_K t}$ separately.

   Each only acts on a few systems (indep of n, constant-sized quantum circuit).

   e.g.



2. But the interesting Hamiltonians have non-commuting summands.

   Idea: suppress and bound the effect of non-commutivity via an appropriate approximation formula.

Trotter formula

For any Hermitian operator A, B and real number t,

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = e^{i(A+B)t} + O\left(\frac{1}{m}\right)$$

Fixing A, B, t, and m is the variable.

Trotter formula

For any Hermitian operator A, B and real number t,

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = e^{i(A+B)t} + O\left(\frac{1}{m}\right)$$

Proof:

$$e^{iAt/m} = I + \frac{i}{m}At + O\left(\frac{1}{m^2}\right), \quad e^{iBt/m} = I + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

power series expansion, test

Trotter formula

For any Hermitian operator A, B and real number t,

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = e^{i(A+B)t} + O\left(\frac{1}{m}\right)$$

Proof:

$$e^{iAt/m} = I + \frac{i}{m}At + O\left(\frac{1}{m^2}\right), \quad e^{iBt/m} = I + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

$$e^{iAt/m}\, e^{iBt/m} = I + \frac{i}{m}At + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

Trotter formula

For any Hermitian operator A, B and real number t,

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = e^{i(A+B)t} + O\left(\frac{1}{m}\right)$$

Proof:

$$e^{iAt/m} = I + \frac{i}{m}At + O\left(\frac{1}{m^2}\right), \quad e^{iBt/m} = I + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

$$e^{iAt/m}\, e^{iBt/m} = I + \frac{i}{m}At + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = \left(I + \frac{i}{m}(A+B)t + O\left(\frac{1}{m^2}\right)\right)^m$$

Trotter formula

For any Hermitian operator A, B and real number t,

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = e^{i(A+B)t} + O\left(\frac{1}{m}\right)$$

Proof:

$$e^{iAt/m} = I + \frac{i}{m}At + O\left(\frac{1}{m^2}\right), \quad e^{iBt/m} = I + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

$$e^{iAt/m}\, e^{iBt/m} = I + \frac{i}{m}At + \frac{i}{m}Bt + O\left(\frac{1}{m^2}\right)$$

$$\left(e^{iAt/m}\, e^{iBt/m}\right)^m = \left(I + \frac{i}{m}(A+B)t + O\left(\frac{1}{m^2}\right)\right)^m$$

binomial expansion

$$= \sum_{k=0}^{m} \binom{m}{k} \left[\frac{i}{m}(A+B)t + O\left(\frac{1}{m^2}\right)\right]^k$$

$$= \sum_{k=0}^{m} \binom{m}{k} \frac{1}{m^k} \left[i(A+B)t + O\left(\frac{1}{m}\right)\right]^k$$

Here $\binom{m}{k} \frac{1}{m^k} = \frac{m!}{k!(m-k)!} \frac{1}{m^k}$

$$= \frac{1}{k!} \frac{m \cdot (m-1) \cdot (m-2) \cdots (m-k+1)}{m \quad m \quad m \quad m}$$

$$= \frac{1}{k!}\left(1 + O\left(\frac{1}{m}\right)\right)$$

Here $\binom{m}{k}\frac{1}{m^k} = \frac{m!}{k!(m-k)!}\,\frac{1}{m^k}$

$$= \frac{1}{k!}\,\frac{m}{m}\cdot\frac{(m-1)}{m}\cdot\frac{(m-2)}{m}\cdots\frac{(m-k+1)}{m}$$

$$= \frac{1}{k!}\left(1+O\left(\frac{1}{m}\right)\right)$$

$$\therefore \left(e^{iAt/m}\,e^{iBt/m}\right)^m = \sum_{k=0}^{m} \frac{1}{k!}\left(1+O\left(\frac{1}{m}\right)\right)\left[i(A+B)t + O\left(\frac{1}{m}\right)\right]^k$$

Here $\binom{m}{k}\frac{1}{m^k} = \frac{m!}{k!(m-k)!}\frac{1}{m^k}$

$$= \frac{1}{k!}\,\frac{m\cdot(m-1)\cdot(m-2)\cdots\cdots(m-k+1)}{m\quad m\quad m\quad m}$$

$$= \frac{1}{k!}\left(1+O\left(\tfrac{1}{m}\right)\right)$$

$$\therefore \left(e^{iAt/m}e^{iBt/m}\right)^m = \sum_{k=0}^{m}\frac{1}{k!}\left(1+O\left(\tfrac{1}{m}\right)\right)\left[i(A+B)t+O\left(\tfrac{1}{m}\right)\right]^k$$

$$= e^{i(A+B)t}\left(1+O\left(\tfrac{1}{m}\right)\right).$$

Here $\binom{m}{k}\frac{1}{m^k} = \frac{m!}{k!(m-k)!}\frac{1}{m^k}$

$$= \frac{1}{k!}\,\frac{m\cdot(m-1)\cdot(m-2)\cdot\cdots\cdot(m-k+1)}{m\quad m\quad m\quad\quad m}$$

$$= \frac{1}{k!}\left(1+O\left(\tfrac{1}{m}\right)\right)$$

$$\therefore \left(e^{iAt/m}e^{iBt/m}\right)^m = \sum_{k=0}^{m} \frac{1}{k!}\left(1+O\left(\tfrac{1}{m}\right)\right)\left[i(A+B)t+O\left(\tfrac{1}{m}\right)\right]^k$$

$$= e^{i(A+B)t}\left(1+O\left(\tfrac{1}{m}\right)\right).$$

Exercise: $\left(e^{iA_1t/m}e^{iA_2t/m}\cdots e^{iA_Lt/m}\right)^m$

$$= e^{i(A+B)t}\left(1+O\left(\tfrac{L}{m}\right)\right).$$

Algorithm for simulating e^{-iHt} on the initial state $|\psi(0)\rangle$ for time T with error $\mathcal{O}(\epsilon)$, where $H = \sum_{k}^{L} H_k$, $\|H_k\| \leq 1$.

Algorithm for simulating e^{-iHt} on the initial state $|\psi(0)\rangle$ for time T with error $\mathcal{O}(\epsilon)$, where $H = \sum_{k}^{L} H_k$, $\|H_k\| \leq 1$.

1. Prepare $|\psi(0)\rangle$ (or an $\mathcal{O}(\epsilon)$ approximation)

Algorithm for simulating e^{-iHt} on the initial state $|\psi(0)\rangle$ for time T with error $\mathcal{O}(\epsilon)$, where $H = \sum_{k}^{L} H_k$, $\|H_k\| \leq 1$.

1. Prepare $|\psi(0)\rangle$ (or an $\mathcal{O}(\epsilon)$ approximation)

2. For j = 1 to L, apply $e^{-iH_jT/m}$.

3. Repeat "step 2" m times.

By the Trotter approximation, we simulated e^{-iHt} within error O(1/m).

Algorithm for simulating e^{-iHt} on the initial state $|\psi(0)\rangle$

for time T with error $O(\epsilon)$, where $H = \sum_k^L H_k$, $\|H_k\| \leq 1$.

1. Prepare $|\psi(0)\rangle$ (or an $O(\epsilon)$ approximation)

2. For j = 1 to L, apply $e^{-iH_jT/m}$.

3. Repeat "step 2" m times.

Choose m = $O\left(\frac{L}{\epsilon}\right)$ gives $e^{-iHT}|\psi(0)\rangle$ with error $\lesssim \epsilon$.

This takes care of approx due to non-commuting H_k's.

Algorithm for simulating e^{-iHt} on the initial state $|\psi(0)\rangle$ for time T with error $O(\epsilon)$, where $H = \sum_k^L H_k$, $\|H_k\| \leq 1$.

1. Prepare $|\psi(0)\rangle$ (or an $O(\epsilon)$ approximation)

2. For j = 1 to L, apply $e^{-iH_jT/m}$.

3. Repeat "step 2" m times.

Choose m = $O\left(\frac{L}{\epsilon}\right)$ gives $e^{-iHT}|\psi(0)\rangle$ with error $\leq \epsilon$.

This takes care of approx due to non-commuting H_k's.

Cost: $e^{-iH_jT/m}$ type of evolution are applied mL times.

Next: the cost due to discrete universal set of gates.

Cost: $e^{-iH_jT/m}$ type of evolution are applied mL times.

Cost due to discrete universal set of gates:

Error for each $e^{-iH_jT/m}$ should be $\leqslant O(\epsilon/mL)$.
Since each H_j acts only on a few systems with constant dim, a circuit of constant #CNOTs and single-qubit gates is sufficient to implement $e^{-iH_jT/m}$.

Cost: $e^{-iH_jT/m}$ type of evolution are applied mL times.

Cost due to discrete universal set of gates:

Error for each $e^{-iH_jT/m}$ should be $\leqslant O(\epsilon/mL)$.
Since each H_j acts only on a few systems with constant dim, a circuit of constant #CNOTs and single-qubit gates is sufficient to implement $e^{-iH_jT/m}$.
The Solovay-Kitaev Thm states that those single qubit gates, with accuracy $O(\epsilon/mL)$, requires polylog(mL/ϵ) gates from the set {H,T}.

So each $e^{-iH_jT/m}$ takes polylog(mL/ϵ) gates.

Cost: $e^{-iH_jT/m}$ type of evolution are applied mL times.

So each $e^{-iH_jT/m}$ takes polylog(mL/ϵ) gates.

Overall circuit has mL polylog(mL/ ϵ) gates.

Cost: $e^{-iH_jT/m}$ type of evolution are applied mL times.

So each $e^{-iH_jT/m}$ takes polylog(mL/ϵ) gates.

Overall circuit has mL polylog(mL/ ϵ) gates.

Finally m = O(L/ϵ), so, a circuit of $O\left(\frac{L^2}{\epsilon}\,\mathrm{poly}\log\frac{L^2}{\epsilon^2}\right)$

gate suffices. (L = poly(n), n = # of subsystems.)

The cost can be much reduced. First, higher order approximation formulae are useful, e.g.,

Baker-Campbell-Hausdorf formula Proof: exercise

$$e^{(A+B)t} = e^{At} \, e^{Bt} \, e^{-\frac{1}{2}[A,B]t^2} + O(t^3)$$

which implies

$$e^{-i(A+B)t} = e^{-iAt/2} \, e^{-iBt} \, e^{-iAt/2} + O(t^3)$$

The cost can be much reduced. First, higher order approximation formulae are useful, e.g.,

Baker-Campbell-Hausdorf formula

Proof: exercise

$$e^{(A+B)t} = e^{At} e^{Bt} e^{-\frac{1}{2}[A,B]t^2} + O(t^3)$$

which implies

$$e^{i(A+B)t} = e^{iAt/2} e^{iBt} e^{iAt/2} + O(t^3)$$

NC Ex 4.49

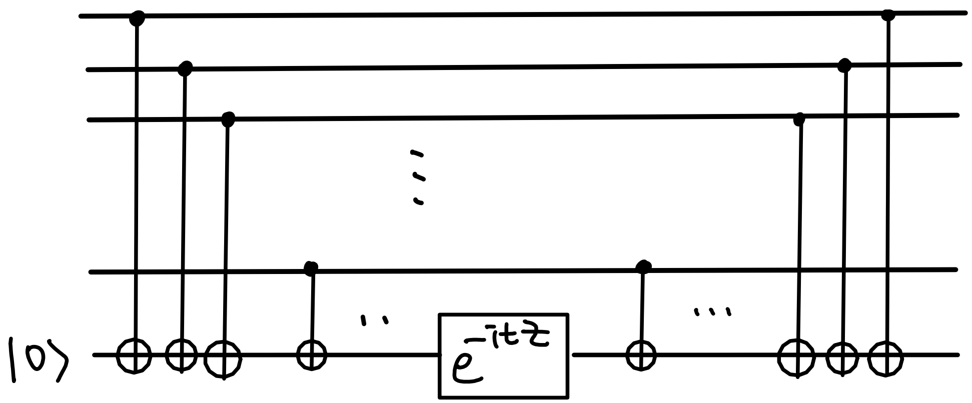
$$e^{-iH_1 t} e^{-iH_2 t} \cdots e^{-iH_L t} e^{-iH_L t} \cdots e^{-iH_2 t} e^{-iH_1 t}$$

$$\simeq e^{-2i(H_1+H_2+\cdots+H_L)t} + O(t^3)$$

Many extensions are known. For example, the H_k need not be local, as long as $e^{-iH_k t}$ can be efficiently implemented (e.g., sparse Hamiltonians)

Some $e^{-iH_k t}$ has much simpler circuits than a decomposition into CNOTs, H, and Ts.

Example:



not necessarily local

$$= e^{-i Z \otimes Z \cdots \otimes Z\, t} \otimes I$$

On the RHS, $Z \otimes Z \otimes \cdots \otimes Z$ is diagonal, and takes

$|x_1\rangle|x_2\rangle \cdots |x_n\rangle$ to $(-1)^{x_1 \oplus x_2 \oplus \cdots \oplus x_n} |x_1\rangle|x_2\rangle \cdots |x_n\rangle$.

e.g., for n=3, $Z \otimes Z \otimes Z = $

$$
\begin{array}{cc}
\begin{bmatrix}
1 & & & & & & & \\
& -1 & & & & & & \\
& & -1 & & & & & \\
& & & 1 & & & & \\
& & & & -1 & & & \\
& & & & & 1 & & \\
& & & & & & 1 & \\
& & & & & & & -1
\end{bmatrix}
&
\begin{array}{c}
000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111
\end{array}
\end{array}
$$

$$
e^{-i Z\otimes Z \otimes Z t} = \begin{bmatrix}
e^{-it} & & & & & & & \\
& e^{+it} & & & & & & \\
& & e^{+it} & & & & & \\
& & & e^{-it} & & & & \\
& & & & e^{+it} & & & \\
& & & & & e^{-it} & & \\
& & & & & & e^{-it} & \\
& & & & & & & e^{+it}
\end{bmatrix}
$$

On the RHS, $Z \otimes Z \otimes \cdots \otimes Z$ is diagonal, and takes

$$|x_1\rangle|x_2\rangle \cdots |x_n\rangle \text{ to } (-1)^{x_1 \oplus x_2 \oplus \cdots \oplus x_n} |x_1\rangle|x_2\rangle \cdots |x_n\rangle .$$

e.g., for n=3,

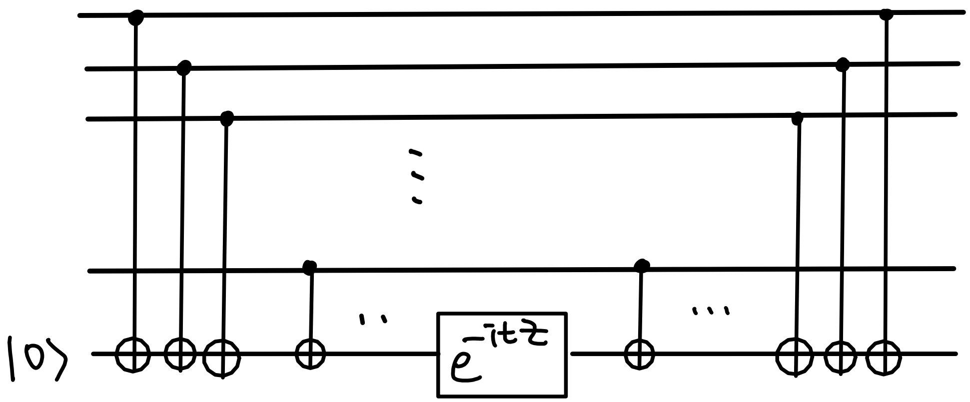
$$Z \otimes Z \otimes Z = \begin{array}{cc} \begin{bmatrix} 1 & & & & & & & \\ & -1 & & & & & & \\ & & -1 & & & & & \\ & & & 1 & & & & \\ & & & & -1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & -1 \end{bmatrix} & \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \end{array}$$

$$e^{-i Z \otimes Z \otimes Z t} = \begin{bmatrix} e^{-it} & & & & & & & \\ & e^{+it} & & & & & & \\ & & e^{+it} & & & & & \\ & & & e^{-it} & & & & \\ & & & & e^{+it} & & & \\ & & & & & e^{-it} & & \\ & & & & & & e^{-it} & \\ & & & & & & & e^{+it} \end{bmatrix}$$

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

takes $|x_1\rangle|x_2\rangle \cdots |x_n\rangle$ to $e^{-i(-1)^{x_1 \oplus x_2 \oplus \cdots \oplus x_n} t} |x_1\rangle|x_2\rangle \cdots |x_n\rangle$.

Example:



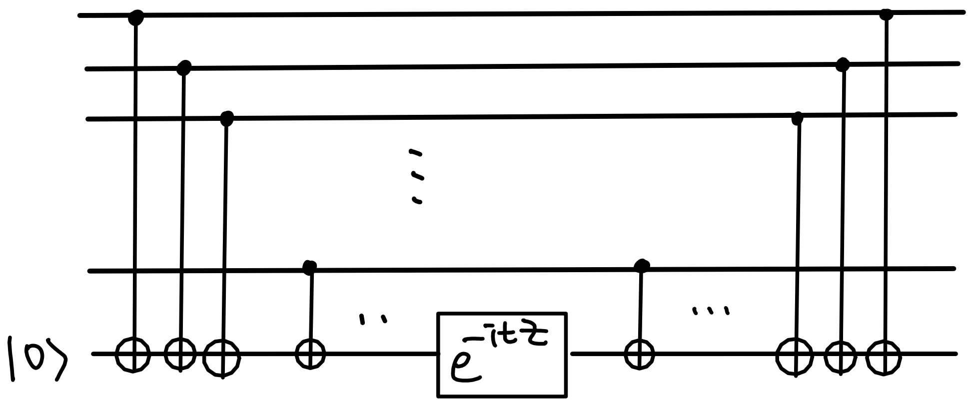
not necessarily local

$$= e^{-i Z\otimes Z \cdots \otimes Z t} \otimes I$$

Proof: On the LHS, the input $|x_1\rangle|x_2\rangle \cdots |x_n\rangle$ evolves as

$$|x_1\rangle|x_2\rangle \cdots |x_n\rangle|0\rangle \xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|x_1 \oplus x_2 \cdots x_n\rangle$$

Example:



not necessarily local

$$= e^{-i Z\otimes Z \cdots \otimes Z\, t} \otimes I$$

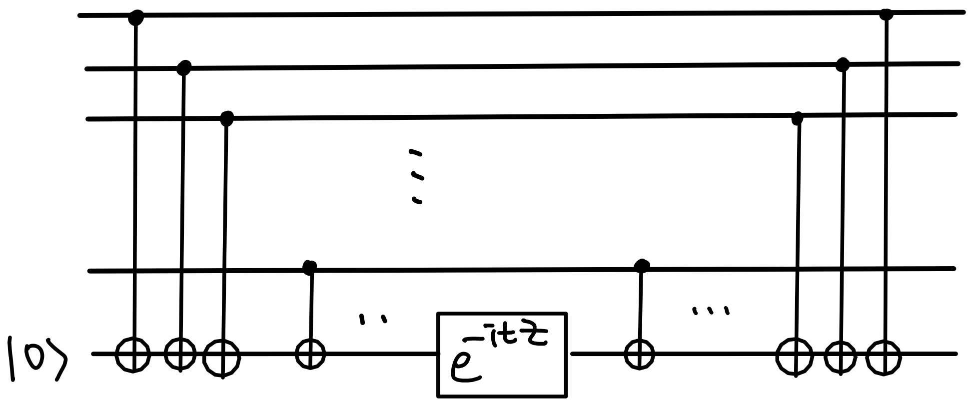
Proof: On the LHS, the input $|x_1\rangle|x_2\rangle \cdots |x_n\rangle$ evolves as

$$|x_1\rangle|x_2\rangle \cdots |x_n\rangle|0\rangle \xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|x_1 \oplus x_2 \cdots x_n\rangle$$

$$\xrightarrow{e^{-iZt}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|x_1 \oplus x_2 \cdots x_n\rangle e^{-i\theta}$$

$$\theta = (-1)^{(x_1 \oplus x_2 \oplus \cdots \oplus x_n)} t$$

Example:



not necessarily local

$$= e^{-i Z \otimes Z \cdots \otimes Z\, t} \otimes I$$

Proof: On the LHS, the input $|x_1\rangle|x_2\rangle \cdots |x_n\rangle$ evolves as

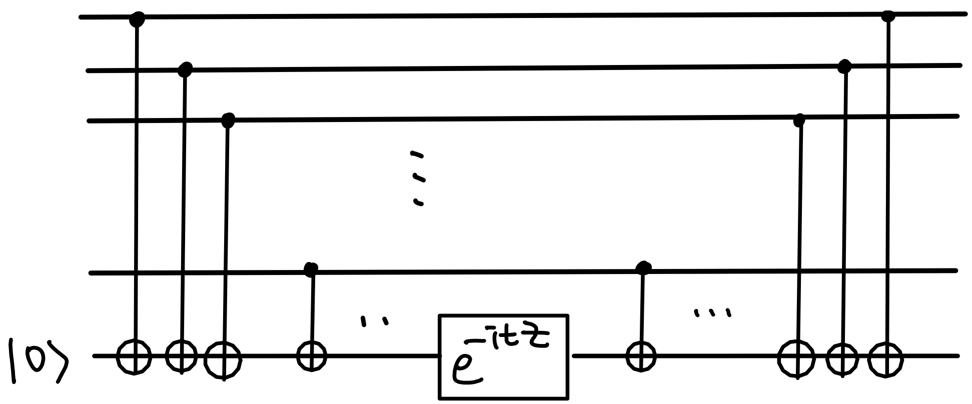
$$|x_1\rangle|x_2\rangle \cdots |x_n\rangle|0\rangle \xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|x_1 \oplus x_2 \cdots x_n\rangle$$

$$\xrightarrow{e^{-iZt}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|x_1 \oplus x_2 \cdots x_n\rangle e^{-i\theta}$$

$$\theta = (-1)^{(x_1 \oplus x_2 \oplus \cdots \oplus x_n)} t$$

$$\xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle \cdots |x_n\rangle|0\rangle e^{-i\theta}$$

Example:



not necessarily local

$$= e^{-i Z\otimes Z \cdots \otimes Z\, t} \otimes I$$

Proof: On the LHS, the input $|x_1\rangle|x_2\rangle\cdots|x_n\rangle$ evolves as

$$|x_1\rangle|x_2\rangle\cdots|x_n\rangle|0\rangle \xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle\cdots|x_n\rangle|x_1\oplus x_2 \cdots x_n\rangle$$

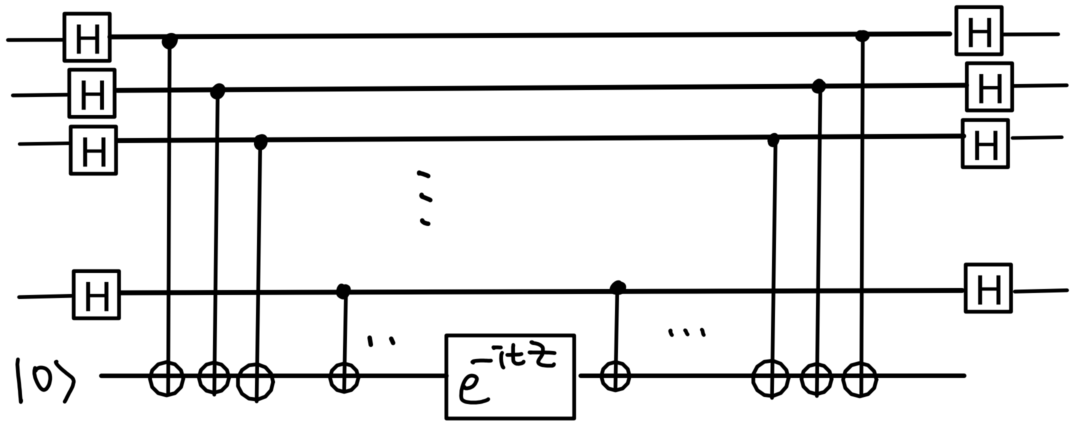
$$\xrightarrow{e^{-iZt}} |x_1\rangle|x_2\rangle\cdots|x_n\rangle|x_1\oplus x_2 \cdots x_n\rangle e^{-i\theta}$$

$$\theta = (-1)^{(x_1\oplus x_2\oplus\cdots\oplus x_n)}\, t$$

$$\xrightarrow{\text{CNOTs}} |x_1\rangle|x_2\rangle\cdots|x_n\rangle|0\rangle e^{-i\theta}$$

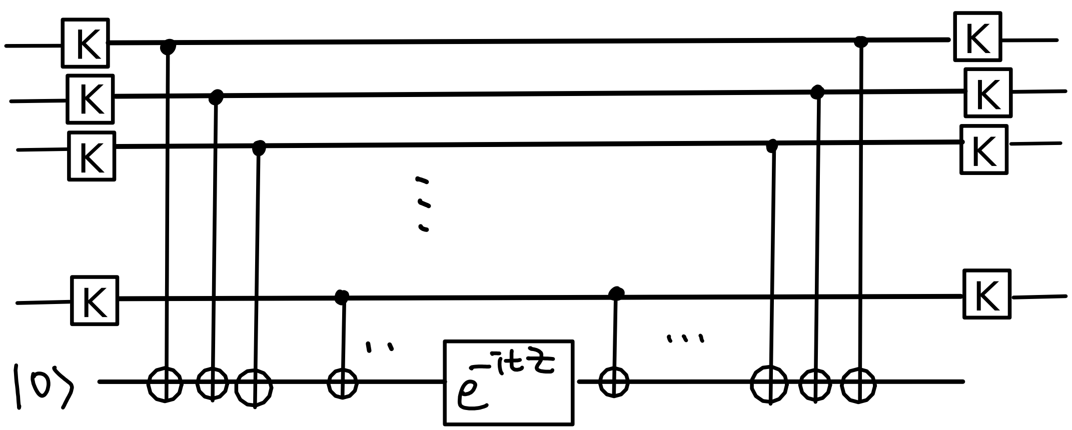
So, LHS = RHS.

Exercise: Show that



$$= e^{-i X\otimes X \cdots \otimes X t} \otimes I$$

and

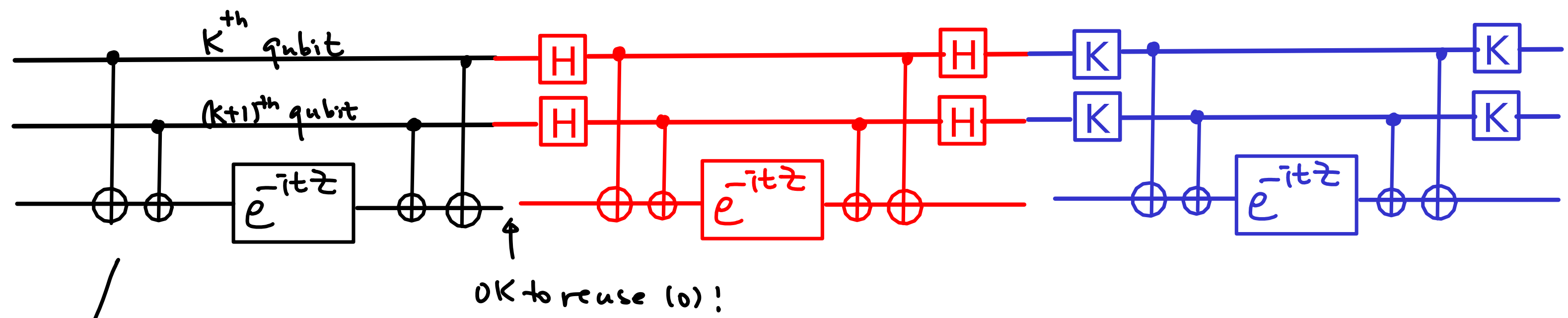


$$= e^{-i Y\otimes Y \cdots \otimes Y t} \otimes I$$

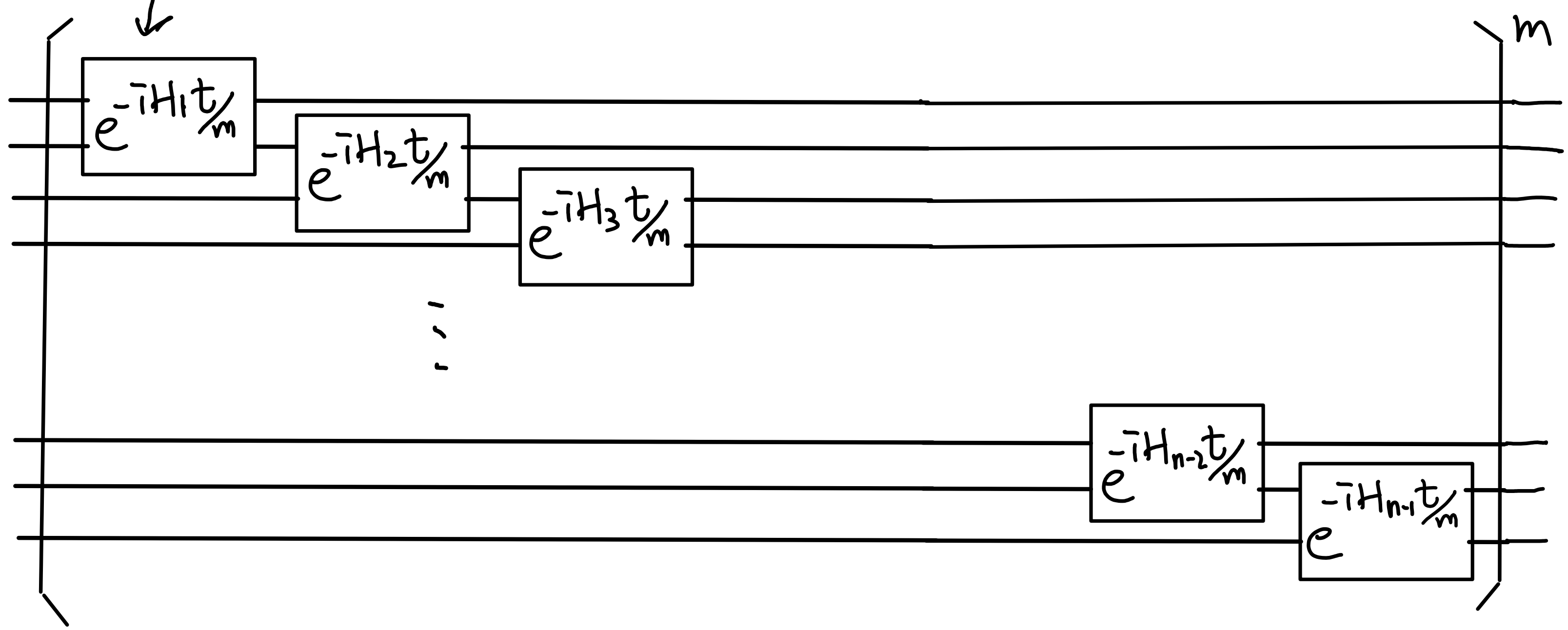
where $K = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -i \\ i & -1 \end{bmatrix}$.

For the Ising model, $H_k = X_k \otimes X_{k+1} + Y_k \otimes Y_{k+1} + Z_k \otimes Z_{k+1}$.

$e^{-iH_k t} =$



To simulate the Hamiltonian $\sum_k H_k$



Interestingly, the possibility to simulate quantum dynamics using a quantum computer also inspires many new efficient classical algorithms to simulate quantum dynamics wherein interesting structures enable such simulations.

e.g., DMRG (density matrix renormalization group)
e.g., MPS (matrix product states)

Even more interestingly, similar inspirations happen in many other areas. e.g., some problems are solved by fast quantum machine learning algorithms, which inspire a similarly efficient classical algorithm.
Reference: google Ewin Tang PhD thesis