Simulating Hamiltonian Dynamics with a Truncated Taylor Series 1412.4687

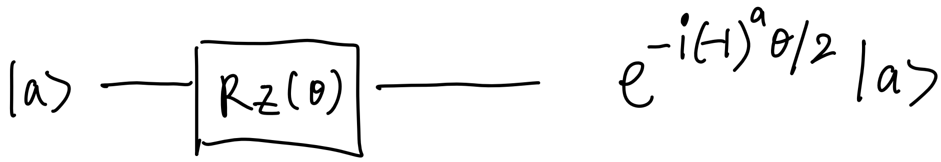
other material: universal quantum simulators. Seth Loyd.

LCU. arxiv: 1202.5822

OAA : 1312.1414.

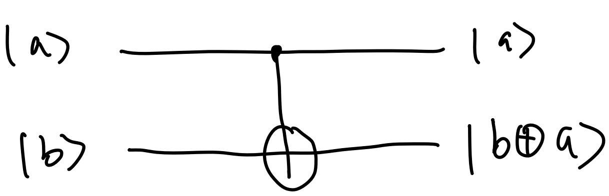
---

① $$R_Z(\theta) = e^{-i\frac{Z\theta}{2}} = \begin{bmatrix} e^{-i\theta/2} & \\ & e^{i\theta/2} \end{bmatrix}$$



② $$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

③ $$CNOT = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{bmatrix}$$



Content.

1. Introduction

Hamiltonian Simulation was first put forward by Richard Feynman. This process is extremely hard for classical computer, of which the dof is bit. It can only take 0/1. However, the dof of QC is qubit. 0./1. superposition of them. One would expect some exponential speed up by utilizing this property. As we have already seen,

Prime factorization is NP hard for C.
P hard for QC.

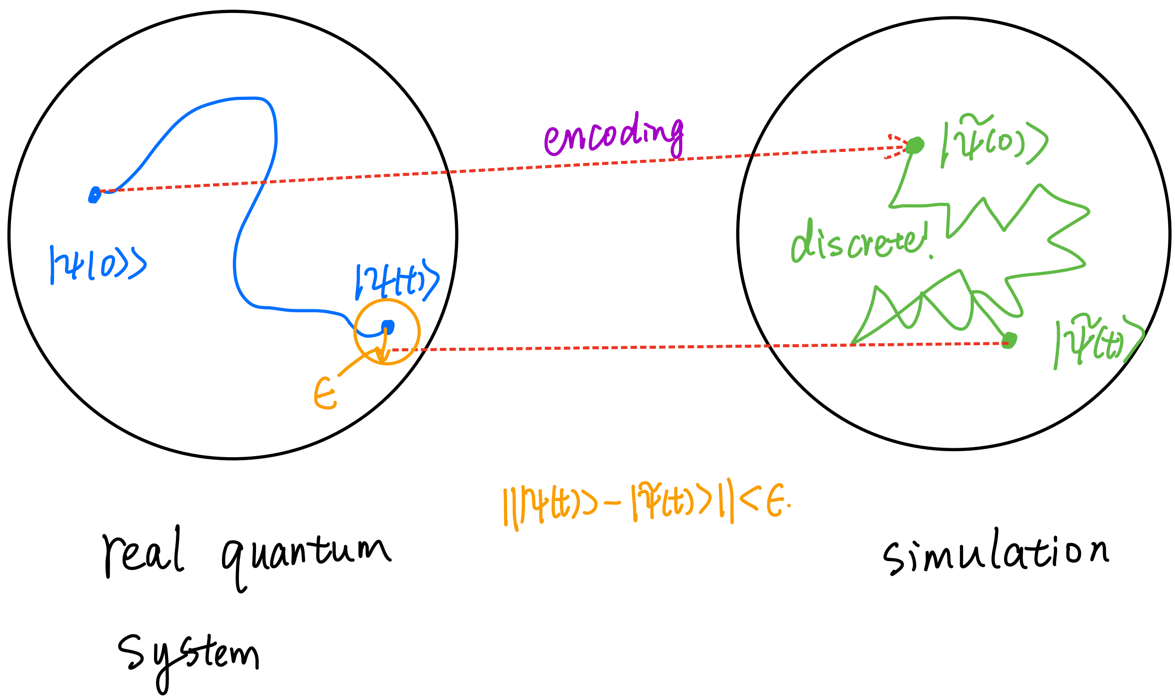
Therefore, we expect to use QC to simulate quantum many-body system to derive the dynamical propers

2、definition

① Hamiltonian simulation refers to the simulation of $U(t) = e^{-iHt}$!

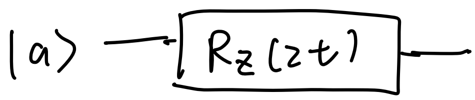
② For a given initial state, we want to predict $|\psi(t)\rangle = U(t)|\psi(0)\rangle$.

A quantum computer can approximate it within error ϵ, $\forall \epsilon > 0$.

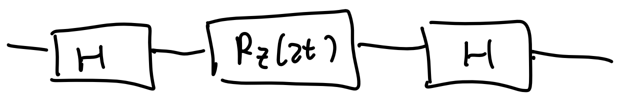


3. Simple Example

① $\hat{H} = Z$, $e^{-iHt} = e^{-iZt}$



② $\hat{H} = X$.

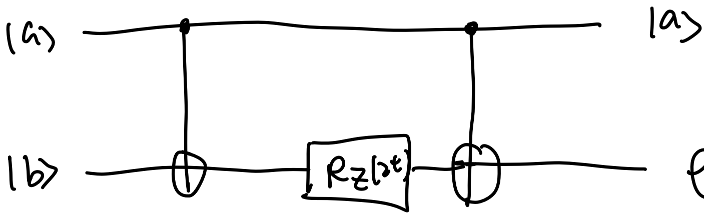


$HZH = X$

③ $\hat{H} = Z \otimes Z$. $e^{-iHt} = e^{-iZ\otimes Zt}$

$$= \cos t\,\mathbb{1} - i\sin t\, Z\otimes Z.$$

$$\begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} \longrightarrow \begin{bmatrix} e^{-it} & & & \\ & e^{it} & & \\ & & e^{it} & \\ & & & e^{-it} \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix}$$] parity.



$e^{-i(-1)^{a\oplus b}t}|b\rangle$

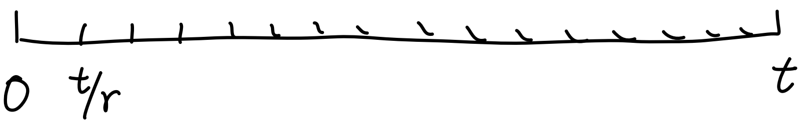
direct product of pauli matrices. can always be simulated.

④ Trotterization : Product of unitary

$$\hat{H} = (\mathbb{1}\otimes X) + Z\otimes Z$$

$$e^{-i(\mathbb{1}\otimes X)t} \quad e^{-iZ\otimes Zt}$$ not commute (but still product of unitary)

Trotter - Suzuki Approximation



$$e^{-i(X\otimes\mathbb{1}+Z\otimes Z)t} \approx \left(e^{-i(X\otimes\mathbb{1})t/r}\, e^{-i(Z\otimes Z)t/r}\right)^r$$

error: $\| \quad V_1 \quad - \quad V_2 \quad \| \in O\left(\frac{t^2}{r}\right)$

$$\frac{t^2}{r} \le \epsilon \quad \therefore \ r \geq t^2/\epsilon .$$

t^2 is too large.

decrease # of gate operation

4. Decomposition & Linear Combination (simulation process)

4.1

Hamiltonian Decomposition n qubit.

$$\hat{H} = \sum_{j=1}^{L=4^n} a_j H_j . \qquad H_j = \underbrace{\sigma_0 \otimes \sigma_1 \otimes \cdots \otimes \sigma_{L-1}}_{n\text{-qubits}} , \quad a_j = \frac{1}{2^n} \mathrm{tr}(H_j H)$$

$\{ I, Z, X, Y \}^{\otimes n}$.

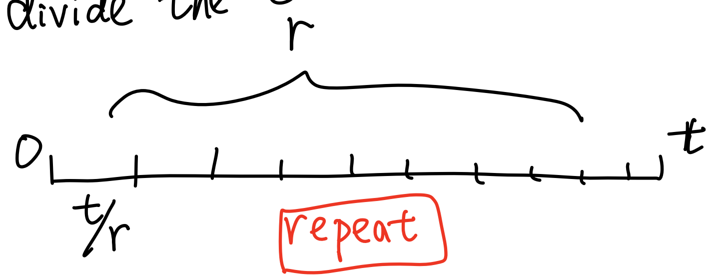
eg. n-qubit

$2^n \times 2^n$, $\mathbb{C}$

4.2

② we want to simulate $U \equiv \exp(-iHt)$ within error ϵ.

③ divide the evolution time into r segments of length t/r



④ error for each segment ϵ/r

$$U = \prod_r U_r$$

$$U_r \equiv \exp(-iHt/r) \approx \sum_{k=0}^{K} \frac{1}{k!} (-iHt/r)^k$$

by tuning r. this can be arbitrary small

$$\| U_r - \tilde{U}_r \| \sim \left\| \frac{1}{(K+1)!} (-iHt/r)^{K+1} \right\| \le \frac{\epsilon}{r}$$

$(K+1)! \sim \sqrt{(K+1)} \left(\frac{K+1}{e}\right)^{K+1}$

$$\Rightarrow K \sim O\left(\frac{\log(r/\epsilon)}{\log\log(r/\epsilon)} \right)$$

⑤ U_r repeat r times. $\Rightarrow U$.

So the key points become simulate U_r.

$$U_r \approx \sum_{k=0}^{K} \sum_{l_1 \cdots l_k = 1}^{L} \frac{(-it/r)^k}{k!} \alpha_{l_1} \cdots \alpha_{l_k} H_{l_1} \cdots H_{l_k} = f(H)$$

$\Rightarrow$ it has a general form.

$\frac{m!}{n!}$

$$\tilde{U} = \sum_{j=1}^{m=L^k} a_j U_j \quad , a_j > 0.$$

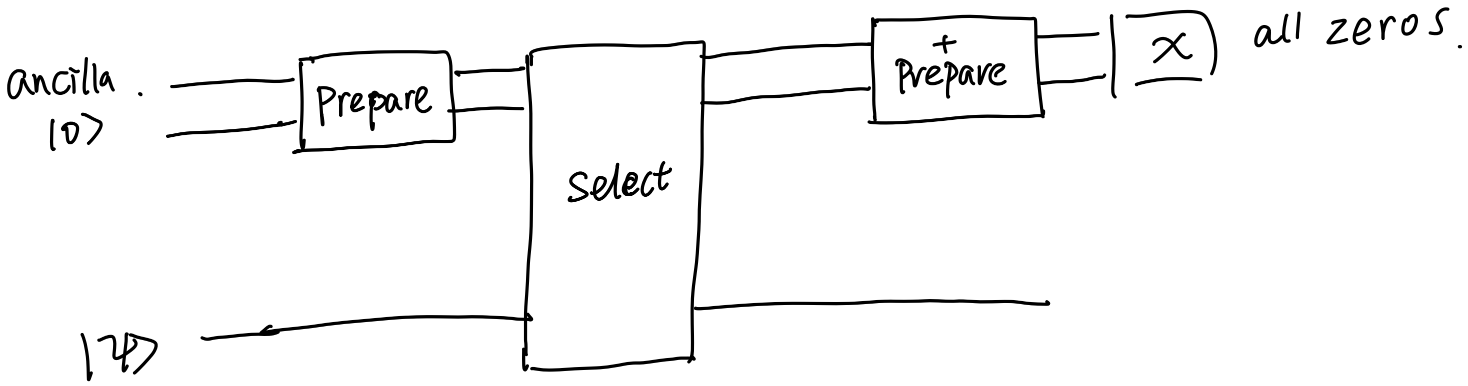
unitary $\propto (-i)^k H_{l_1} H_{l_2} \cdots H_{l_k}$.

$$a = \sum_j a_j = \sum_{k}\sum_{L} \frac{(t/r)^k}{k!} \alpha_{l_1} \cdots \alpha_{l_k}$$

$\geq (\min\{\alpha_{l_i}\}) \cdot e^{t/r}$

$a_j \sim O(e^{t/r})$

$$\boxed{\tilde{U}|\psi\rangle}$$



1. First step.

prepare ancillas: $|0\rangle \longrightarrow \sum_{j=1}^{L^K} \frac{\sqrt{a_j}}{\sqrt{a}} |j\rangle$, $a=\sum a_j$.

∵ we "always" have way to do this. for example.

$|0\rangle \xrightarrow[H]{} \frac{|0\rangle+|1\rangle}{2}$.

# of ancilla: $\log L^K = K \log L, \propto \frac{\log L \log(t/\epsilon)}{\log\log(t/\epsilon)}$

2. Second Step. apply select operator.

$[\text{select}]|j\rangle|\psi\rangle \longrightarrow |j\rangle U_j|\psi\rangle$.

eg. select $= \sum_j |j\rangle\langle j| \otimes U_j$

$$|0\rangle|\psi\rangle \xrightarrow{\text{prep.}} \sum_j \frac{\sqrt{a_j}}{\sqrt{a}}|j\rangle|\psi\rangle$$

$$\xrightarrow{\text{select}} \sum_j \frac{\sqrt{a_j}}{\sqrt{a}}|j\rangle U_j|\psi\rangle$$

$$\xrightarrow{\text{prep}^\dagger} \text{prepare}^\dagger \sum_j \frac{\sqrt{a_j}}{\sqrt{a}}|j\rangle U_j|\psi\rangle \equiv |\chi\rangle$$

$$\xrightarrow{\text{measure.}} |0\rangle\langle 0| \otimes \mathbb{I}\, |\chi\rangle.$$

$$= \frac{1}{a}|0\rangle \tilde{U}|\psi\rangle$$

$$\tilde{U} = \sum a_j U_j.$$

$$\text{prepare}\,|0\rangle = \sum_j \frac{\sqrt{a_j}}{\sqrt{a}}|j\rangle$$

$$\langle 0|\,\text{prepare}^\dagger = \sum_j \frac{\sqrt{a_j}}{\sqrt{a}}\langle j|$$

Reflect: The trick of LCU. is block Encoding. By introducing additional degree of freedom, we can select the U_j

$$\begin{array}{c|c} & \begin{array}{cc} |0\rangle & \quad |1\rangle \end{array} \\ \begin{array}{c} |0\rangle \\ \\ |1\rangle \end{array} & \begin{bmatrix} [\tilde{U}/a] & ? \\ & \\ ? & ? \end{bmatrix} \end{array}$$

$\rightarrow$ project it back to the $|0\rangle\langle 0|$ span.

But how about the projection probability/successful probability.

$$P_{succ} = \frac{\langle \psi | \tilde{U}^\dagger \tilde{U} | \psi \rangle}{a^2}.$$

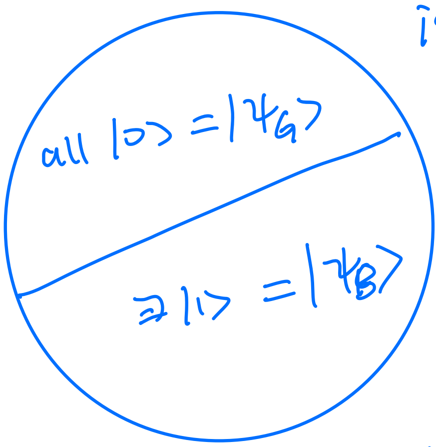
$$\propto \frac{1}{a^2} \approx e^{-2t/r}$$

$$P_{succ}^{r} \sim e^{-2t}$$

The problem comes from Block Encoding.
The exponentially small probability comes from the fact that not all the auxilla qubits. are in $|0\rangle$ state.

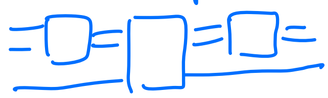
Therefore we need to use some method to boost the probability.

5. Amplitude Amplification.

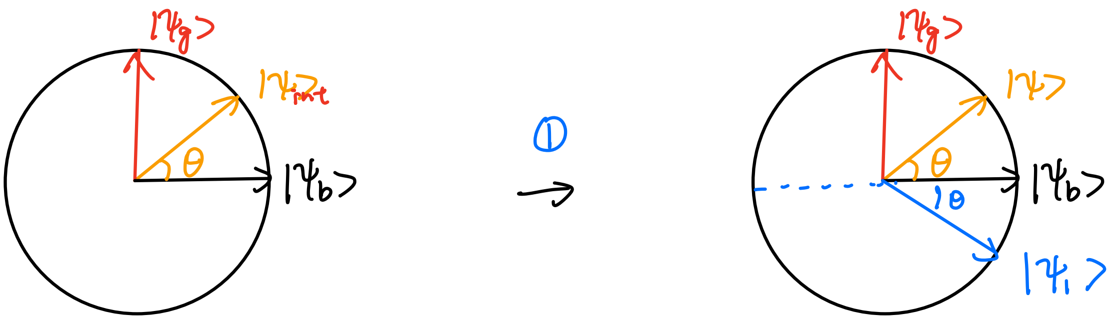


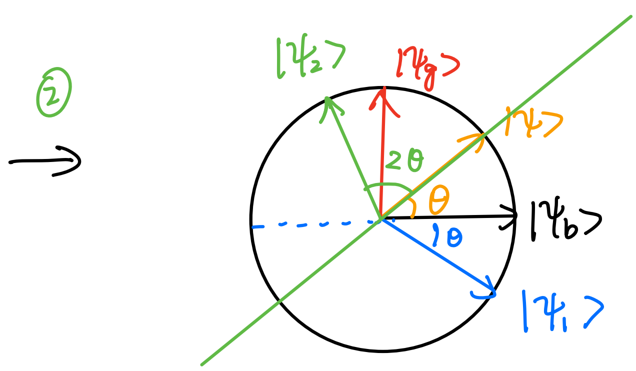
So the way we boost the possibility is to rotate our state back to the $|0\rangle\langle 0|$ Hillbert space.

$|\psi_{int}\rangle = |x\rangle$



$$W = -(I - |\psi_{int}\rangle\langle\psi_{int}|)(\mathbb{1} - P_G)$$





$$W = \boxed{-(\mathbb{I} - 2|\psi_{int}\rangle\langle\psi_{int}|)}\boxed{(1 - 2|\psi_g\rangle\langle\psi_g|)}$$

$P_G = |0\rangle\langle 0|^{\otimes m} \otimes \mathbb{I}.$

$P_B = \mathbb{I} - |0\rangle\langle 0|^{\otimes m} \otimes \mathbb{I}$

$$①② = W = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

W^m: repeat m times. map $\begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} \rightarrow \begin{bmatrix} \sin(2m+1)\theta \\ \cos(2m+1)\theta \end{bmatrix}$

we want $(2m+1) = \frac{\pi}{2}$

$\#$ of W: $m = \frac{1}{2}\left(\frac{\pi}{2\theta} - 1\right)$

if initial $P = \frac{1}{4} \Leftrightarrow \theta = \frac{\pi}{6}$, then $m = 1$.

6. Gate Counts

circuit depth : t accuracy: ϵ

Technique	Gate complexity	Query complexity
Product formula 1st order	$O\left(\frac{t^2}{\epsilon}\right)$ [7] 1996	$O\left(d^3 t \left(\frac{dt}{\epsilon}\right)^{\frac{1}{2}k}\right)$ [9]
Taylor series	$O\left(\frac{t\log^2\left(\frac{t}{\epsilon}\right)}{\log\log\frac{t}{\epsilon}}\right)$ [7] 2014	$O\left(\frac{d^2\|\|H\|\|_{\max}\log\frac{d^2\|\|H\|\|_{max}}{\epsilon}}{\log\log\frac{d^2\|\|H\|\|_{\max}}{\epsilon}}\right)$ [6]
Quantum walk	$O\left(\frac{t}{\sqrt{\epsilon}}\right)$ [7] 2012	$O\left(d\|\|H\|\|_{\max}\frac{t}{\sqrt{\epsilon}}\right)$ [5]
Quantum signal processing	$O\left(t+\log\frac{1}{\epsilon}\right)$ [7] 2017	$O\left(td\|\|H\|\|_{\max}+\frac{\log\frac{1}{\epsilon}}{\log\log\frac{1}{\epsilon}}\right)$ [8]

Where $||H||_{\max}$ is the largest entry of H.

classical bits $\propto 2^n$

No fast Forwarding Theorem

prepare :

$$U_r \approx \sum_{k=0}^{K} \sum_{l_1 \cdots l_k = 1}^{L} \underbrace{\frac{(-it/r)^k}{k!}}_{K} \underbrace{\alpha_{l_1} \cdots \alpha_{l_k}}_{L} H_{l_1} \cdots H_{l_k}$$

$$O(KL) \sim O\left(L \frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\right)$$