Jogging a Hamiltonian

Arjo, Rutvij

May 4, 2022

Contents

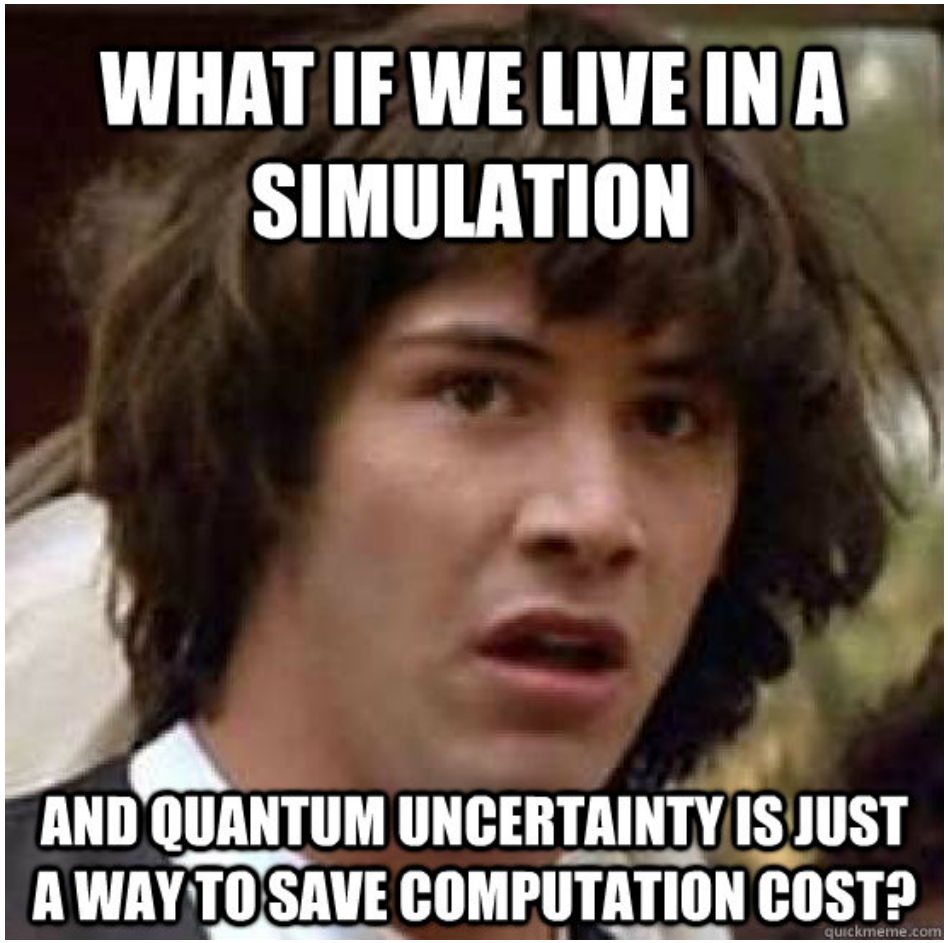
Introduction

The problem of efficiently simulating a Hamiltonian is one of the long standing problems in computational physics. Today, the problem of Hamiltonian simulation stands as one of the most impacting and significant contribution of quantum computers. Furthermore since it is a BQP-complete problem, devising any efficient classical algorithm for the same is believed to be intractable.

In this project, we plan to study and analyse Hamiltoninan simulation using quantum walks. We will also compare the efficiency of these algorithms with other methods such as trotterization and quantum signal processing.

Introduction (totally not a filler)

Sorry for abstracting so much in the first slide! Here, is a meme for the pains endured.



Applications of Hamiltonian Simulation

- Drug design and protein folding: Finding the native structure of the protein is equivalent to the problem of finding the ground state of the system.
- Graph theoretic problems: Several graph theory related problems such as Graph Coloring (which is NP-complete) can be cleverly mapped to finding ground states of some classes of Hamiltonians.
- Note that finding the ground state of an Hamiltonian is equivalent to finding the min $\langle\phi| H |\phi\rangle , \forall \ |\phi\rangle$.
- Furthermore, Hamiltonian simulation can be used as a subroutine for implementing continuous-time quantum walks and solving linear equations, amongst others.

Time-independent Hamiltonians

A Hamiltonian can be time-independent (then you are lucky) or time-dependent.

- Time-independent: $U(t) = e^{-iHt}$
- Time-dependent: $U(t) = e^{-i\int_0^t H(s)ds}$

In this project, we will be dealing with sparse time-independent Hamiltonians.

Local Hamiltonians

Here, H denotes a local hamiltonian where each H_j acts on $k = O(1)$ qubits.

$$H = \sum_{j=1}^{n} H_j$$

- A k-local Hamiltonian with n terms is d-sparse with $d = 2^k n$.
- A k-local Hamiltonian with n terms can be expressed as a linear combination of $\leq n4^k$ Pauli operators, each of which are unitaries.

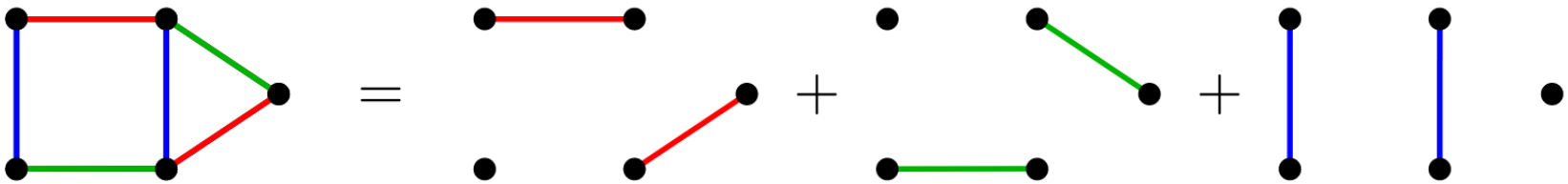
Sparse Hamiltonians

Consider the matrix form of the Hamiltonian, H, to be $N \times N$. Then, H is said to be sparse if it has at most d non zero entries per row $d = \text{poly}(\log N)$.

Note: There should exist some way for efficiently computing the location and value of the j^{th} nonzero entry $\forall\ r \in$ rows of H. This is called row computability.

Problem of efficient Decomposition

- Finding the efficient decomposition of a given Hamiltonian is not an easy task.
- One approach for dealing with the same for a sparse Hamiltonian (H) is by representing H in terms of a graph and having its edges colored.



Sparse Hamiltonian Simulation

- **Trotterization**: This was described by Zeeshan, Shreyas and Hrishi, in quite some detail.
- **Truncated Taylor Series**: We shall describe this algorithm in some depth, before moving on with quantum walks.
- **Quantum Walks**: The main focus of our project which we describe below in quite some detail.
- **Quantum Signal Processing**: This procedure gives us the best known theoretical bounds through a framework called qubitization.

Complexity Comparision

Method	*Query Complexity*	*Gate Complexity*
Lee-Suzuki Trotter	$O\left(d^3 t \left(\frac{dt}{\epsilon}\right)^{\frac{1}{2}k}\right)$	$O\left(\frac{t^2}{\epsilon}\right)$
Truncated Taylor Series	$O\left(\frac{d^2 \|\|H\|\|_{max} \log \frac{d^2 \|\|H\|\|_{max}}{\epsilon}}{\log\log \frac{d^2 \|\|H\|\|_{max}}{\epsilon}}\right)$	$O\left(\frac{t \log\left(\frac{t}{\epsilon}\right)}{\log\log \frac{t}{\epsilon}}\right)$
Quantum Walks	$O\left(d\|\|H\|\|_{max} \frac{t}{\sqrt{\epsilon}}\right)$	$O\left(\frac{t}{\sqrt{\epsilon}}\right)$
Qubitisation	$O\left(td\|\|H\|\|_{max} + \log \frac{1}{\epsilon}\right)$	$O\left(t + \log \frac{1}{\epsilon}\right)$

Complexity Comparision

1. Is there a lower bound on sparse Hamiltonian Simulation?
2. Can we simulate the evolution of a Hamiltonian till time t faster than $O(t)$?

These questions are answered by the $H_{PARITY \circ OR} \otimes H_{COMPLETE}$ and the no-fast-forwarding theorem.

Complexity Comparision

Takeaways:

- Lower bound on query complexity for sparse Hamiltonian simulation $= \Omega(t\sqrt{d}||H||_{1\to 2})$.
- Here, $||H||_{1\to 2} = \max_j \sqrt{\sum_i |H_{ij}|^2}$.
- Any physically realistic Hamiltonian (which can be efficiently simulated by a quantum circuit) is highly unlikely.
- Here, fast forwarding is the ability to simulate (using a quantum computer) the evolution of a given system governed by a certain Hamiltonian H to within time t, such that the simulation takes time which is much less than t (sub-linear time simulation).

Future Questions

How well do the bounds reflect upon implementation over NISQ frameworks?

We will talk about this at the end, since by then we will have a better understanding of the subroutines associated with each Hamiltonian simulation technique.

Future Questions

What if the Hamiltonian is not sparse?

If a Hamiltonian is not sparse then, we cannot jot down the nonzero entries of a row in polynomial time - hence it will no longer be efficiently row computable.

In that case, we would again have to tackle with the problem of efficient decomposition, albeit in a different and arguably harder setting.

Truncated Taylor Series

Simulating Hamiltonians using truncated Taylor series is based on the principle of an Hamiltonian being decomposible as a linear combination of unitaries.

- Let H be some given Hamiltonian.
- $H = \sum_{j=1}^{n} \alpha_j H_j$
- Here, each H_j is unitary and a mechanism is available for implementing that unitary.
- Local Hamiltonians: These can be decomposed into a sum of tensor products of Pauli matrices where each term acts nontrivially on a constant number of qubits.
- We can use this algorithm, broadly, for all families of sparse Hamiltonians.

Truncated Taylor Series

- Divide the evolution time into r segments.
- Within each segment we have $U_r = e^{-iHt/r} \approx \sum_{k=0}^{K} \frac{1}{k!}(-iHt/r)^k$, where the Taylor series is truncated to K.
- For U_r, the corresponding error should be $\leq \epsilon/r$.
- Given, $r \geq ||H||t$, we can choose $K = O(\frac{\log(r/\epsilon)}{\log\log(r/\epsilon)})$.
- Overall complexity $\approx rK$.

Truncated Taylor Series

$$
\begin{aligned}
U_r &= \sum_{k=0}^{K} \frac{(-iHt/r)^k}{k!} \\
&= \sum_{k=0}^{K} \sum_{l_1, l_2, \ldots, l_k = 1}^{L} \frac{(t/r)^k}{k!} \alpha_{l1}, \ldots, \alpha_{lk} (-i)^k H_{l1} \ldots H_{lk} \\
&= \sum_{j=0}^{J} \beta_j V_j
\end{aligned}
$$

Truncated Taylor Series

Here, U_r is a LCU of $V_j = (-i)^k H_{l1} \dots H_{lk}$ with the coefficients β_j described above. Now, without loss of generality we can set each of $\alpha_l \geq 0 \implies \beta_j > 0$. Note that, $U_r = \tilde{U}(t/r)$ where $\tilde{U}$ denotes truncation.

$$||\tilde{U}(t/r) - U(t/r)|| \leq e^{\alpha t/r} \frac{(\alpha t/r)^{K+1}}{(K+1)!}$$

Now, here we have the following as well.

- $\alpha = \sum_l \alpha_l$
- Given ϵ, $K = O\left(\frac{\log \frac{\alpha t}{\epsilon}}{\log \log \frac{\alpha t}{\epsilon}}\right)$
- Required complexity $= O(rK)$

Truncated Taylor Series

Implementation: We have described the associated implementation in our project report.

Future work: Coming up with faster and less error prone algorithms by further exploiting commutative and anti-commutative relations within the framework for truncated Taylor series.

Another question: How general is this technique? Can it be used even for Time-dependent Hamiltonians?

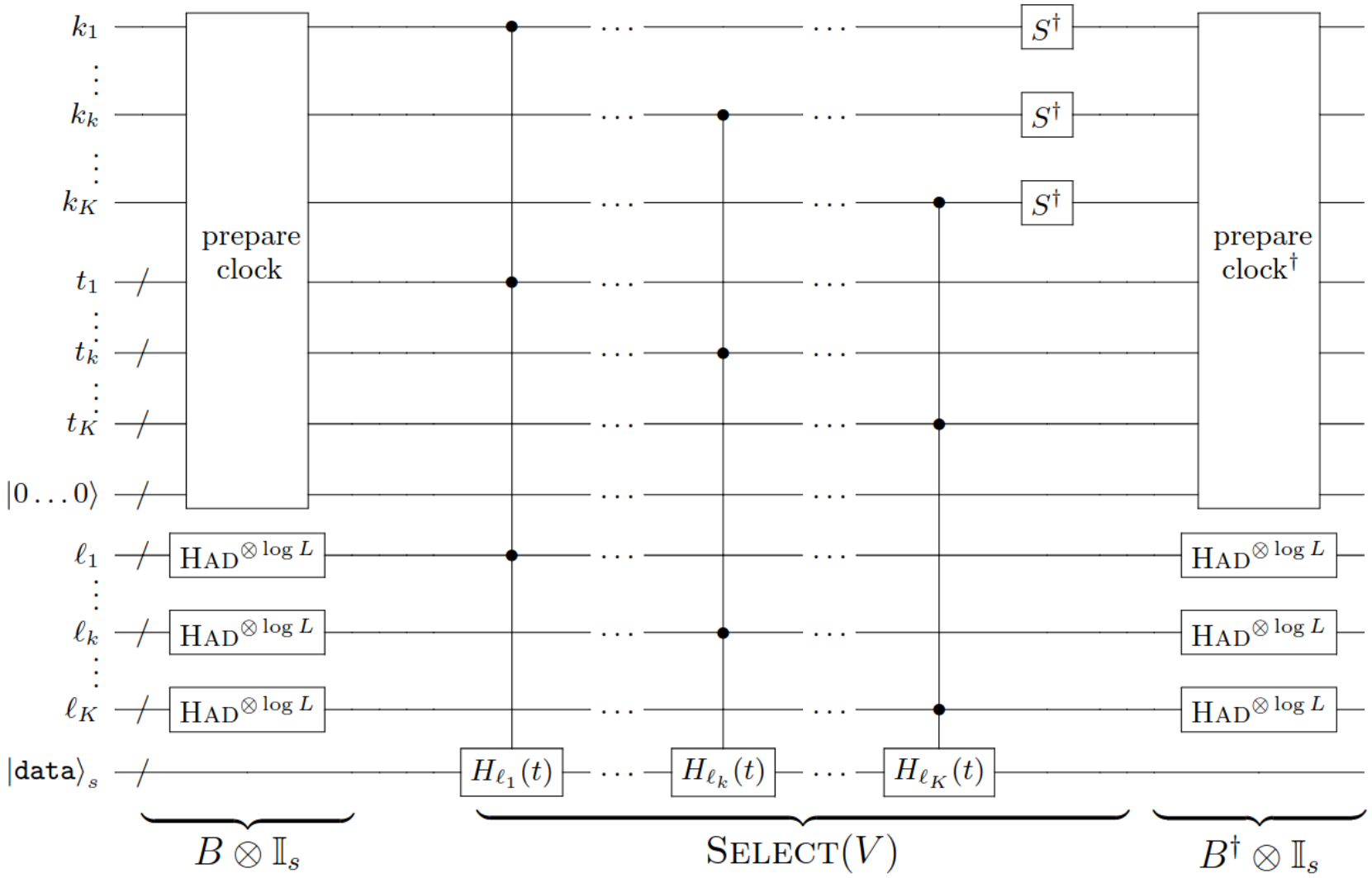
Time-dependent Hamiltonian

We observe time dependent Hamiltonians when our system has external classical control on it. then, we can not represent the corresponding unitary evolution as a closed form expression. Thus, U remains as following.

$$U(t) = e^{-i \int_0^t H(s)}$$

For this time-dependent case, we do have an infinite-series representation - an analog for Taylor series in the time-independent case. This representation is called the Dyson series expression.

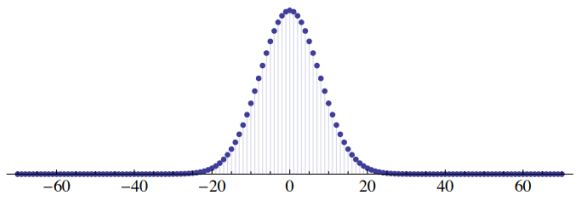
Circuit for truncated Infinite Series Sum



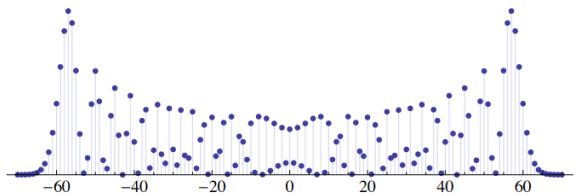
Quantum Walks

Quantum walks are the quantum mechanical counterpart of classical random walks with randomness arising through:

1. Quantum superposition of states,
2. Non-random, reversible unitary evolution
3. Collapse of the wave function due to state measurements.



classical



quantum

Quantum Walks

There are two kinds of quantum walks: discrete and continuous quantum walks. The main difference between these two sets is the timing used to apply corresponding evolution operators.

- **Discrete Quantum Walks**: The evolution operator $\hat{U}$ is applied only in discrete time steps. The mathematical structure of this model is evolution via unitary operator, i.e.

$$|\psi\rangle_{t_2} = \hat{U}\,|\psi\rangle_{t_1}$$

- **Continuous Quantum Walks**: The evolution (Hamiltonian) operator H of the system can be applied with no timing restrictions. The mathematical structure of this model is evolution via the Schrödinger equation, i.e.

$$|\psi\rangle_t = e^{-iHt}\,|\psi\rangle_0$$

Szegedy Quantum Walks

We first define $|d\rangle = \sum_{j=1}^{N} d_j |j\rangle$ to be the principal eigenvector of abs(H), and then we define N orthonormal quantum states $|\psi_1\rangle, \ldots, |\psi_N\rangle \in \mathbb{C}^N \times \mathbb{C}^N$ as

$$|\psi_j\rangle = \frac{1}{\sqrt{||\text{abs}(H)||}} \sum_{k=1}^{N} \sqrt{H_{jk}^* \frac{d_k}{d_j}} |j,k\rangle$$

Then we define the swap operator S such that

$$S|j,k\rangle = |k,j\rangle$$

and the isometry mapping

$$T = \sum_{j=1}^{N} |\psi_j\rangle \langle j|$$

Szegedy Quantum Walks

Childs defines the discrete-time quantum walk (Szegedy) corresponding to H as the unitary operator obtained by first reflecting about span$\{|\psi_j\rangle\}$ and then exchanging the two registers with the swap operator S. Thus, the walk operator is defined as

$$U = iS(2TT^\dagger - \mathbb{I})$$

For $\frac{H}{||\text{abs}(H)||}|\lambda\rangle = \lambda|\lambda\rangle$, the eigenvalues and eigenvectors of the walk operator U are

$$|\mu_\pm\rangle = \frac{\mathbb{I} - e^{\pm i \arccos\lambda} S}{\sqrt{2(1-\lambda^2)}} T|\lambda\rangle, \mu_\pm = \pm e^{\pm i \arcsin\lambda}$$

Simulation using Quantum Walks

$$T|\lambda\rangle = \frac{1-\lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}|\mu_+\rangle + \frac{1-\lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}|\mu_-\rangle$$

To simulate the given Hamiltonia H using the corresponding Szegedy discrete-time quantum walk operator U, a phase of $e^{i\lambda t}$ is to be introduced for the λ term of the superposition observed in the above expression. The procedure used for simulation is:

1. Apply T to the input state $|\psi\rangle$
2. Perform phase estimation of U, estimating a phase $\pm e^{\pm i\arcsin\lambda}$ for the component $T|\lambda\rangle$
3. Use estimate of $\arcsin\lambda$ to get estimate of λ, $\tilde{\lambda}$.
4. Induce the phase $e^{-i\tilde{\lambda}t}$ and uncompute $\tilde{\lambda}$ by performing phase estimation in reverse.
5. Finally apply reverse isometry $T^\dagger$

Simulation using Quantum Walks

For an eigenstate-eigenvalue pair $|\theta\rangle, e^{i\theta}$ of U, we define P denoting the isometry that performs phase estimation on this walk as

$$P|\theta\rangle = \sum_{\phi} a_{\phi|\theta} |\theta, \phi\rangle$$

where $a_{\phi|\theta}$ is the amplitude for the estimate ϕ. And let F_t is the unitary operation that applies the desired phase to a given value-estimate state. That is,

$$F_t |\theta, \phi\rangle = e^{-it \sin \phi} |\theta, \phi\rangle$$

Childs claims that the simulation of Hamiltonian evolution e^{iHt} is $T^\dagger P^\dagger F_t P T$, that is

$$e^{iHt} \approx T^\dagger P^\dagger F_t P T$$

Simulation Using Quantum Walks (Math Explained)

For $|\psi\rangle \in \mathbb{C}^N$ such that $|\psi\rangle = \sum_\lambda \psi_\lambda |\lambda\rangle$

$$T|\psi\rangle = \sum_\lambda \psi_\lambda \left(\frac{1 - \lambda e^{-i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_+\rangle + \frac{1 - \lambda e^{i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_-\rangle \right)$$

$$PT|\psi\rangle = \sum_\lambda \psi_\lambda \left(\sum_\phi a_{\phi|\arcsin\lambda} \frac{1 - \lambda e^{-i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_+\rangle \right.$$
$$\left. + a_{\phi|\pi-\arcsin\lambda} \frac{1 - \lambda e^{i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_-\rangle \right)$$

$$F_t PT|\psi\rangle = \sum_\lambda \psi_\lambda e^{-it\sin\phi} \left(\sum_\phi a_{\phi|\arcsin\lambda} \frac{1 - \lambda e^{-i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_+\rangle \right.$$
$$\left. + a_{\phi|\pi-\arcsin\lambda} \frac{1 - \lambda e^{i \arccos\lambda}}{\sqrt{2(1-\lambda^2)}} |\mu_-\rangle \right)$$

Simulation Using Quantum Walks (Math Explained)

We note that ϕ is the phase estimation for $\arcsin\lambda$, and thus $\sin\phi = \tilde{\lambda}$.

$$P^\dagger F_t P T \left|\psi\right\rangle = \sum_\lambda \psi_\lambda e^{-it\tilde{\lambda}} \left(\frac{1 - \lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}} \left|\mu_+\right\rangle + \frac{1 - \lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}} \left|\mu_-\right\rangle \right)$$

$$T^\dagger P^\dagger F_t P T \left|\psi\right\rangle = \sum_\lambda \psi_\lambda e^{-it\tilde{\lambda}} \left|\lambda\right\rangle$$

Thus,

$$T^\dagger P^\dagger F_t P T = \sum_\lambda e^{-it\tilde{\lambda}} \left|\lambda\right\rangle\left\langle\lambda\right| \approx \sum_\lambda e^{-it\lambda} \left|\lambda\right\rangle\left\langle\lambda\right| = e^{-iHt}$$

Simulation Using Quantum Walks (Complexity Analysis)

$$
\begin{aligned}
|\left\langle \psi | e^{iHt} T^\dagger P^\dagger F_t P T | \psi \right\rangle| &= \left| \sum_{\lambda,\phi} |\psi_\lambda|^2 e^{i(\lambda - \sin\phi)t} \left(\frac{\left|a_{\phi|\arcsin\lambda}\right|^2}{2} \right.\right. \\
&\qquad \left.\left. + \frac{\left|a_{\phi|\pi-\arcsin\lambda}\right|^2}{2} \right) \right| \\
&\geq 1 - \frac{t^2}{2} \min_{\theta\in[0,2\pi)} \sum_{\phi} (\theta - \phi)^2 |a_{\phi|\theta}|^2
\end{aligned}
$$

We use the following as initial state for phase estimation instead of equal superposition

$$
\sqrt{\frac{2}{M+1}} \sum_{x=0}^{M-1} \sin\left(\frac{\pi(x+1)}{M+1}\right) |x\rangle
$$

Simulation Using Quantum Walks (Complexity Analysis)

The probability distribution of estimates of θ with this initial state is

$$\left|a_{\theta+\Delta|\theta}\right|^2 = \frac{\cos^2\left(\Delta\frac{M+1}{2}\right)\sin^2\left(\frac{\pi}{M+1}\right)}{2M(M+1)\sin^2\left(\frac{\Delta}{2}+\frac{\pi}{2(M+1)}\right)\sin^2\left(\frac{\Delta}{2}-\frac{\pi}{2(M+1)}\right)}$$

where the estimated phase is $\frac{2\pi j}{M} = \theta + \Delta$ for some integer j. A range of angles can be chosen such that $\Delta = \Delta_0 + \frac{2\pi j}{M}$, where $0 \leq \Delta_0 \leq \frac{2\pi}{M}$ and $-\lceil\frac{M}{2}\rceil + 1 \leq j \leq \lfloor\frac{M}{2}\rfloor$.
For sufficiently large M, we get that the bound on the fidelity of the simulation is atleast $1 - 93t^2/M^2$. Hence, the complexity of simulation is

$$O(t/\sqrt{\delta})$$

Qubitization: a failed attempt at understanding (?)

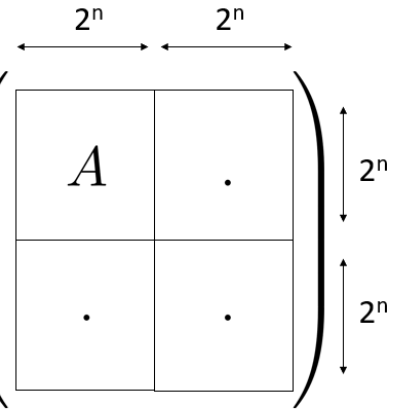
Qubitization provides us with a general framework to understand a high-dimensional space as the direct sum of a large number of 2-dimensional spaces on which we act simultaneously and without their interfering with each other.

Qubitization is found to perform better than older Hamiltonian simulation techniques for most cases. But, its use cases are not constrained to the simulation of Hamiltonians. It can be applied to understand and discover new algorithms and properties especially in the field of quantum walks.

Qubitization

- Qubitization is based on the block encoding technique.

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}$$



- It relies on the decomposition of the 2^{n+1} dimensional space on which U acts into 2^n spaces, each of dimension 2.
- Each such space of dimension 2 is generated by an eigenvector of A and another associated vector.

Qubitization

- Thus, qubitization refers to this decomposition in spaces of dimension 2.
- Each eigenspace of A is immersed in a space of dimension 2, i.e. in the qubit space.
- Let $\tilde{U}$ be the result of qubitizing U.

$$\tilde{U} = \oplus_{\lambda \in \Lambda(A)} W_{\text{qubit},\lambda}$$

Thus, $\tilde{U}$ acts as the direct sum of 2^n unitary matrices of dimension 2. We have one unitary of dimension 2 per eigenvalue of A.

Qubitization

- For the purpose of Hamiltonian simulation, A will be some Hamiltonian, H.
- $W_{\text{qubit},\lambda}$ is a 2×2 diagonalizable unit matrix. Hence, $\forall\ \lambda_W \in \Lambda(W),\ |\lambda_W| = 1$. Furthermore, λ_Ws are determined by $\lambda \in \Lambda(H)$. W is referred to as the walk operator.

$$W_{\text{qubit},\lambda} \approx \begin{bmatrix} e^{i \cos^{-1} \lambda} & 0 \\ 0 & e^{-i \cos^{-1} \lambda} \end{bmatrix}$$

- Now, the next step involves transforming the eigenvalues of the walk operator from $e^{\pm i \cos^1 \lambda}$ to $e^{i\lambda t}$.

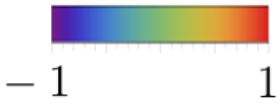
Qubitization

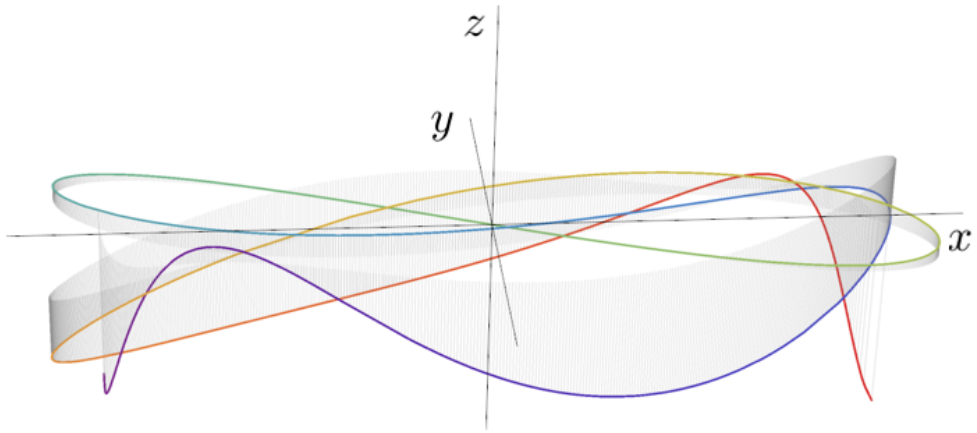
- This transformation is possible by using any QSVT method (in the given diagram, we can see that QSP has been used).
- This method is suitable for finding static quantities such as the ground state of the associated H (Hamiltonian).
- For estimating ground state, we can just apply QPE directly to W.
- **Associated Query Complexity**: $O(t + \log(1/\epsilon))$

Qubitization

$$\begin{pmatrix} H & \cdots \\ \vdots & \ddots \end{pmatrix} \xrightarrow[\text{Qubitization}]{} e^{\pm i \sin^{-1}[H]} \xrightarrow[\text{Quantum signal processing}]{} \begin{pmatrix} f[H] & \cdots \\ \vdots & \ddots \end{pmatrix}$$

Input eigenvalue λ





Output eigenvalue $\sqrt{1-|\vec{x}_\lambda|^2} + i x_\lambda$

A Fake Conclusion

Qubitization (however it may magically work) is the best.

Comparing Implementability (as a conclusion)

Similar to implementing truncated Taylor series, qubitization also involves two subroutines within its framework of implementation.

Now unlike as in Trotterization, these subroutines are quantum in nature (and not classical).

Thus, they require ancilla qubits (to be exact, logarithmically more qubits when compared to Trotterization) and hence might be infeasible for useful implementations in extremely near term NISQ-era.