JOGGING A HAMILTONIAN:

Simulating Hamiltonians with several methods

RUTVIJ [1] AND ARJO [2]

Apr 28, 2022

CONTENTS

[1] *Rutvij Menavlikar*, CQST, IIITH

[2] *Alapan Chaudhuri*, CQST, IIITH

ABSTRACT

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.

1 INTRODUCTION

Hamiltonian simulation serves as the basis for many problems in science. The universe is of quantum nature, which means that to simulate particles or any other natural entities you need to simulate its Hamiltonian. However, computing the evolution given the Hamiltonian is also not very straightforward.

Due to the discrete nature of classical computers, it is not possible to efficiently simulate any of the above on the same. This gives quantum computers an edge since they can work with qubits to simulate the dynamics of a system given its Hamiltonian. This makes Hamiltonian Simulation the most impacting and significant contribution of quantum computers.

Furthermore, simulating hamiltonian is a BQP-complete problem. Thus, it is believed to be intractable by classical computers. Devising any efficient classical algorithm for hamiltonian simulation would mean proving that $P = BQP$.

1.1 Why simulate Hamiltonians?

Computational simulation of the physical systems provides important insights and acts as a bridge between theory and experiments.

Moreover, it has more than a handful of applications, namely:

- 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.

1.2 Why simulate Hamiltonians in quantum computers?

Computational simulation of the physical systems provides important insights and acts as a bridge between theory and experiments. But can a quantum system be efficiently simulated by a classical computer? The answer is certainly (almost), ‘No!’ as said by Bell. Even simulation using pseudo-random variables has exponential computational overhead.

In the case where the Hamiltonian consists of a sum of interaction terms between small subsystems, the simulation is thought to be exponentially more efficient than classical simulation.

2 THE HAMILTONIAN

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

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

2.1 Local Hamiltonians

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

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

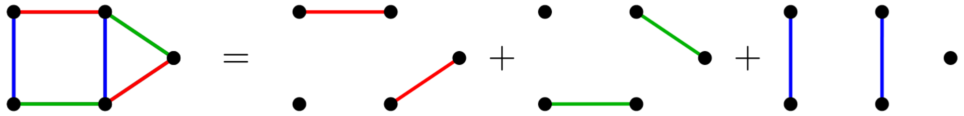
2.2 Sparse Hamiltonians

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 $d = \text{poly}(\log N)$. Some interesting points on sparse Hamiltonians:

- 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.
- 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 $\leqslant n4^k$ Pauli operators, each of which are unitaries.

2.3 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. Now, given that a sparse graph can be efficiently colored using only local information, we can conclude that this procedure will result in a decomposition yielding efficient simulations. In layman terms, the simulation breaks into small pieces that are easy to handle (shown below diagramatically).



A rough sketch of pre-computation involving decomposition before simulating a sparse Hamiltonian.

- Decompose the d-sparse Hamiltonian into a sum of terms each with eigenvalues 0 and π (up to an overall shift and rescaling).

- Perform edge-coloring.
- Approximately decompose into terms with all non-zero entries equal.
- Remove zero blocks so that all terms have two fixed eigenvalues.

3 ALGORITHMS FOR SPARSE HAMILTONIAN SIMULATION

Today, there exists several frameworks for simulating Hamiltonians, namely:

- **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.

Now, for implementation purposes over near term NISQ models of quantum computation, Trotterization appears to be the way to go, even though it is theoretically slower with respect to algorithms such as qubitization. This is because it is simpler to implement and uses way less ancilla qubits.

Now, it is also worth noting that theoretically Trotterization in itself is not inefficient. On the contrary, it is quite efficient achieving a complexity only slightly superlinear in t. Why is that good? Well, because the no-fast-forwarding theorem implies that simulation in sublinear time is not possible.

3.0.1 *No-fast-forwarding Theorem*

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, but such that the simulation takes time which is much less than t.

The no-fast-forwarding theorem states that fast forwarding any physically realistic Hamiltonian (which can be efficiently simulated by a quantum circuit) is highly unlikely. This is proven by showing that even for 2-sparse row-computable Hamiltonians the possibility of exponential fast-forwarding implies $BQP = PSPACE$.

3.1 Complexity comparision for sparse Hamiltonian simulation

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	$O\left(\frac{d^2 \lVert H\rVert_{max} \log \frac{d^2 \lVert H\rVert_{max}}{\epsilon}}{\log\log \frac{d^2 \lVert H\rVert_{max}}{\epsilon}}\right)$	$O\left(\frac{t \log\left(\frac{t}{\epsilon}\right)}{\log\log \frac{t}{\epsilon}}\right)$
Quantum Walks	$O\left(d \lVert H\rVert_{max} \frac{t}{\sqrt{\epsilon}}\right)$	$O\left(\frac{t}{\sqrt{\epsilon}}\right)$
Qubitisation	$O\left(td \lVert H\rVert_{max} + \log \frac{1}{\epsilon}\right)$	$O\left(t + \log \frac{1}{\epsilon}\right)$

3.2 Unsolved questions

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

2. Quantum walks serve as a framework for universal computation. In that case, can we leverage Hamiltonian simulation using quantum walks to solve unrelated graph theoretic and combinatorial optimization problems?

3. The problem of efficient decomposition is required to be tackled for general cases.

4. Given possible frameworks for Hamiltonian simulation and a given Hamiltonian with a favoured decomposition, which framework will be most suitable for simulating it? Can this problem be solved using decision trees/random forests by keeping calculating information gained about the original Hamiltonian by studying how it is deemed to affect chosen sets of qubits?

5. How to deal with non-sparse Hamiltonians?

4 TRUNCATED TAYLOR SERIES

Any Hamiltonian can be decomposed as a linear combination of unitary matrices. Now, let H be some given Hamiltonian. Then, we have the following:

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

4.1 The Simulation

Consider $U(t) = e^{-iHt}$ and we wish to simulate this evolution within error ϵ.

- 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 $\leqslant \epsilon/r$.
- Given, $r \geqslant ||H||t$, we can choose $K = O(\frac{\log(r/\epsilon)}{\log\log(r/\epsilon)})$.
- Overall complexity $\approx rK$.

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

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 $\alpha_l \geqslant 0 \implies \beta_j > 0$. Note that, $U_r = \tilde{U}(t/r)$ where $\tilde{U}$ denotes truncation.

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

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 = $O(rK)$

4.2 Implementation

In order to be able to implement the given sum of unitary operators, we introduce a few subroutines. The first subroutine we introduce is the SELECT such that $\text{SELECT}(V) \left|j\right\rangle \left|\psi\right\rangle = \left|j\right\rangle V_j \left|\psi\right\rangle$. Thus, SELECT serves as some form of reflection.

Now, in order to simulate $\tilde{U}$, we first introduce a $J+1$ dimensional unitary B to prepare the required ancillary state (where $s = \sum_{j=0}^{J} \beta_j$).

$$B\left|0\right\rangle = \frac{1}{\sqrt{s}} \sum_{j=0}^{J} \sqrt{\beta_j} \left|j\right\rangle$$

Now, let us define the operator $W = (B^\dagger \otimes \mathbb{I})\, \text{SELECT}(V)(B \otimes \mathbb{I})$ then we have the following for some $\left|\phi\right\rangle$ whose ancillary state is supported in the subspace orthogonal to $\left|0\right\rangle$.

$$W\left|0\right\rangle \left|\psi\right\rangle = \frac{1}{s} \left|0\right\rangle \tilde{U} \left|\psi\right\rangle + \sqrt{1 - \frac{1}{s^2}} \left|\phi\right\rangle \implies P(W\left|0\right\rangle \left|\psi\right\rangle) = \frac{1}{s} \left|0\right\rangle \tilde{U} \left|\psi\right\rangle$$

Here, P denoted the projection operator $\left|0\right\rangle \left\langle 0\right| \otimes \mathbb{I}$. The value of s can be adjusted by choosing the size of the segments. Now, using the *Approximate segment lemma* and *Oblivious amplitude amplification*, we obtain that if $\tilde{U}$ is unitary then it can be implemented using W, $W^\dagger$ and with the $(J+1)$-dimensional ancilla reflection $R = 1 - 2P$.

To conclude, if $\tilde{U}$ is unitary and $A = -WRW^\dagger RW$ then, $A\left|0\right\rangle \left|\psi\right\rangle = \left|0\right\rangle \tilde{U} \left|\psi\right\rangle$.

4.3 Possible future work

Now, it is worth noting that in this formulation, the truncation error exists even if the Hamiltonian consists of mutually commutative terms. Thus, we can ideally come up with faster and less error prone algorithms by further exploiting commutative and anti-commutative relations.

5 TIME-DEPENDENT HAMILTONIANS

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.

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

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.

5.1 Dyson expansion of U

$$U(t) = 1 + (-i)\int_0^t dt_1 H(t_1) + (-i)^2 \int_0^t \int_0^t {}_1 dt_1 dt_2 H(t_1)H(t_2) + \dots$$

Note that, in this case ordering highly matters because when we are considering H(t), it may not always commute with itself (because it is time dependent).

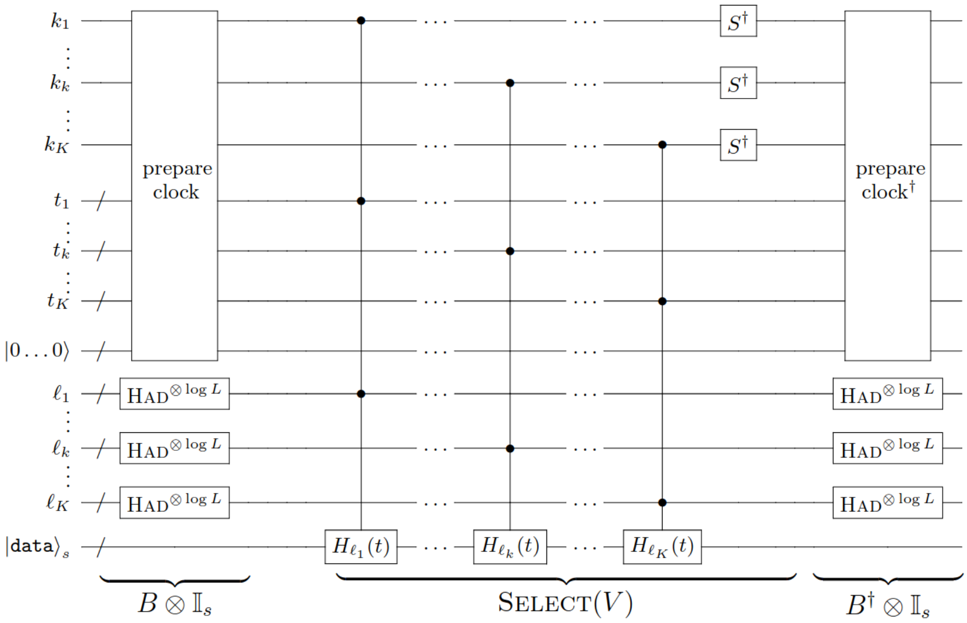


Figure 1: Truncated Dyson series implementation for time-dependent Hamiltonian simulation

6 QUANTUM WALKS

Quantum walks, the quantum mechanical counterpart of classical random walks are an advanced tool for building quantum algorithms that have been recently shown to constitute a universal model of quantum computation. In contrast to the classical random walk, where the walker occupies definite states and the randomness arises due to stochastic transitions between states, in quantum walks randomness arises through:

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

The quantum dynamics of any discrete system can be captured by a quantum walk on a network, which is a universal model for quantum computation. Besides being a useful primitive to design quantum algorithms, quantum walks are a powerful tool to model transport in quantum systems such as the transfer of excitations in light-harvesting systems. Studying the long-time dynamics of quantum walks on networks is crucial to the understanding of these diverse problems.

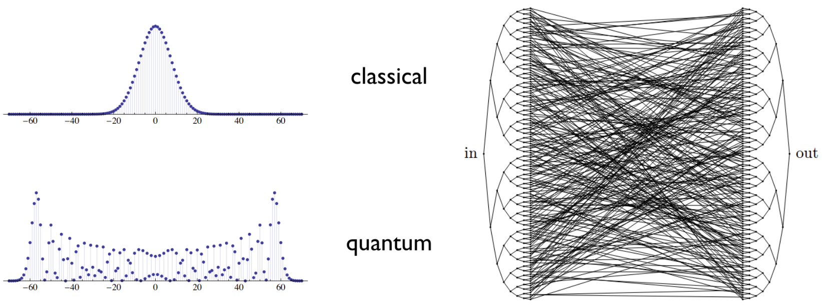


Figure 2: Quantum vs Classical Random Walks

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**: It consists of two quantum mechanical systems, named a walker and a coin (for random element selection), as well as an evolution operator which is applied to both systems 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**: consists of a walker and an evolution (Hamiltonian) operator of the system that can be applied with no timing restrictions at all, i.e. the walker walks any time. The mathematical structure of this model is evolution via the Schrödinger equation, i.e. $|\psi\rangle_t = e^{-\frac{iHt}{\hbar}}\,|\psi\rangle_0$

6.0.1 Defining a Discrete Quantum Walk on a Graph

Let $G = (V, E)$ be a $d-$regular graph with $|V|$=n, and $\mathcal{H}_v$ be the Hilbert space spanned by states $|v\rangle$ where $v \in V$. We also define $\mathcal{H}_A$, which is the coin space, as an auxiliary Hilbert space of dimension d spanned by the basis state $\{|i\rangle\,|i \in \{1, \ldots, d\}\}$, and $\hat{C}$, which is the coin operator and a unitary transformation on $\mathcal{H}_A$. We also define a shift operator $\hat{S}$ on $\mathcal{H}_v \otimes \mathcal{H}_A$ such that $\hat{S}\,|a, v\rangle = |a, u\rangle$ where u is the a^{th} neighbour of v.

Finally we define one step of the quantum walk on G as $\hat{U} = \hat{S}\left(\hat{C} \otimes \hat{I}\right)$. And if the quantum walk starts with an initial state $|\psi\rangle_0$, then the state after t steps can be expressed as $|\psi\rangle_t = \hat{U}^t\,|\psi\rangle_0$

6.0.2 Defining a Continuous Quantum Walk on a Graph

Let $G = (V, E)$ be a graph with $|V|$=n. We first define a generator matrix M of order n given by

$$M_{ab} = \begin{cases} -\gamma, & a \neq b, (a, b) \in G \\ 0, & a \neq b, (a, b) \notin G \\ k\gamma, & a = b \text{ and } k \text{ is the degree of vertex } a \text{ in } G \end{cases}$$

Now we define a Hamiltonian $\hat{H}$ with elements from matrix M in a Hilbert space $\mathcal{H}$ with basis $\{|1\rangle, |2\rangle, \ldots, |n\rangle\}$. That is,

$$\left\langle a|\hat{H}|b\right\rangle = M_{ab}$$

Now for a quantum walk starting in initial state $|\psi\rangle_0$, then the state after time t can be expressed as

$$|\psi\rangle_t = e^{-i\hat{H}t}\,|\psi\rangle_0$$

6.1 Szegedy Quantum Walk

Szegedy defined a discrete-time quantum walk corresponding to an arbitrary discrete-time classical Markov chain in [1]. Based on his work, Andrew Childs described a discrete-time quantum walk corresponding to a general N × N Hermitian matrix H.

First, Childs fixed an orthonormal basis $\{|j\rangle : j = 1, 2, \ldots, N\}$ of $\mathbb{C}^N$, and

$$\text{abs}(H) = \sum_{j,k=1}^{N} \left|H_{jk}\right| |j\rangle \langle k|$$

is the elementwise absolute value of H. Now we define the principal eigenvector of abs(H), i.e., the eigenvector with eigenvalue $||\text{abs}(H)||$ to be

$$|d\rangle = \sum_{j=1}^{N} d_j |j\rangle$$

Now Childs defines N orthonormal quantum states $|\psi_1\rangle, \ldots, |\psi_N\rangle \in \mathbb{C}^N \times \mathbb{C}^N$ as

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

where the state $\left|\psi_j\right\rangle$ is similar to a state $|j\rangle \otimes \left|\phi_j\right\rangle$ if H represented a graph with nodes $\{|1\rangle, \ldots, |N\rangle\}$ and $|\phi_k\rangle = \sum_{k=1}^{N} a_k |k\rangle$ is the superposition of all the neighbours of $|j\rangle$ and a_k is the amplitude corresponding to the edge weight of the edge (j, k) (as H_{jk} corresponds to the edge weight in an adjancency matrix).

6.1.1 Swap Operator

The function of the swap operator S is

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

Thus, $S = S^\dagger$ and $SS^\dagger = S^2 = \mathbb{I}$.

6.1.2 Discrete Time Quantum Walk

Childs defines the discrete-time quantum walk (Szegedy) corresponding to H as the unitary operator obtained by first reflecting about $\text{span}\{\left|\psi_j\right\rangle\}$ and then exchanging the two registers with the swap operator S. To formally define this, we first define the isometry mapping from $|j\rangle \in \mathbb{C}^N$ to $\left|\psi_j\right\rangle \in \mathbb{C}^N \times \mathbb{C}^N$ as

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

Then $TT^\dagger$ is the projector onto $\text{span}\{\left|\psi_j\right\rangle\}$. And thus the walk operator is defined as $U = iS(2TT^\dagger - \mathbb{I})$. Now we consider the eigenvalues and eigenvectors for $\frac{H}{||\text{abs}(H)||}$,

$$\frac{H}{||\text{abs}(H)||} |\lambda\rangle = \lambda |\lambda\rangle$$

We get the eigenvalues and eigenvectors of the walk operator U to be

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

7 HAMILTONIAN SIMULATION USING QUANTUM WALK

No-fast-forwarding theorem states that the optimal number of steps required to simulate an evolution for time t is linear in t. This can be done using Szegedy Quantum Walks and phase estimation.

Before moving on to the procedure for Hamiltonian simulation, we observe that for any state $|\psi\rangle \in \mathbb{C}^N \times \mathbb{C}^N$, we can write $T|\psi\rangle$ as

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

To simulate H using the corresponding 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 $T^\dagger$

Now for an eigenstate-eigenvalue pair $|\theta\rangle, e^{i\theta}$ of the discrete-time quantum walk, let P denote the isometry that performs phase estimation on this walk. Then P can be defined as follows

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

where $a_{\phi|\theta}$ is the amplitude for the estimate ϕ. And let F_t be 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 $T^\dagger P^\dagger F_t P T$. To prove this claim, Child derived the inner product between the expected evolved state, $e^{-iHt}|\psi\rangle$, and the simulated evolved state, $T^\dagger P^\dagger F_t P T|\psi\rangle$ to be

$$\left\langle \psi | e^{iHt} T^\dagger P^\dagger F_t P T | \psi \right\rangle = \sum_{\lambda,\phi} |\psi_\lambda|^2 e^{i(\lambda - \sin\phi)t} \frac{\left|a_{\phi|\arcsin\lambda}\right|^2 + \left|a_{\phi|\pi-\arcsin\lambda}\right|^2}{2}$$

He then went on to derive lower bound for the fidelity of this simulation to be

$$\left| \left\langle \psi | e^{iHt} T^\dagger P^\dagger F_t P T | \psi \right\rangle \right| \geqslant 1 - \frac{t^2}{2} \min_{\theta\in[0,2\pi)} \sum_{\phi} (\theta - \phi)^2 |a_{\phi|\theta}|^2$$

To further quantify the fidelity of the simulation, evaluation of performance of phase estimation is required. In standard phase estimation modulo M, we begin from the state of equal superposition, that is $\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle$. Although there are some other states that, starting from whom give better results. One particular state that Childs mentioned is

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

He claims, and goes on to prove that this initial state minimizes the variance of the estimate and gives the best bound for the fidelity of the simulation.

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 $0 \leqslant \Delta_0 \leqslant \frac{2\pi}{M}$ and $-\lceil\frac{M}{2}\rceil + 1 \leqslant j \leqslant \lfloor\frac{M}{2}\rfloor$.

With these assumptions, and for sufficiently large M, we get that the bound on the fidelity of the simulation is atleast $1 - 93t^2/M^2$. Thus we find that $M = O(t/\sqrt{\delta})$ steps suffice to obtain fidelity at least $1-\delta$. Hence, the gate complexity of simulation is also

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

With K applications of the operator U for phase estimation of $\arcsin\lambda$, the variance is approximately $\left(\frac{\pi}{K}\right)^2$, which means the variance in estimate of λ is $O\left(\frac{||H||_1^2}{2K^2}\right)$ and that translates to error being $O\left(\frac{||H||_1 t}{K}\right)$. Taking $= 1$, to achieve atleast $1-\delta$ fidelity, $K = O\left(\frac{||H||_1 t}{\sqrt{\delta}}\right)$. Hence, the query complexity of simulation is

$$O\left(\frac{d||H||_{max} t}{\sqrt{\delta}}\right)$$

Note that $||H||_1 \leqslant d||H||_{max}$ as the Hamiltonian is given to be d-sparse. Also, we use this upper bound for complexity as $||H||_1$ is often unknown.

8 QUBITIZATION

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.

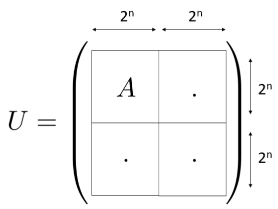
Qubitization is found to perform better than older Hamiltonian simulation techniques for most cases. But, its use cases are not constrained to the simulation of Hamiltonians. It can be applied to understand and discover new algorithms and properties especially in the field of quantum walks.

8.1 Description

Before we delve into qubitization, we should note that in quantum computing, embedding in a 2-dimensional space is not computationally expensive. A single auxiliary qubit is enough to double the dimension of the vector space used to represent the quantum information being processed. This is way treating qubits classically results in dealing with exponential space since we require a vector space of dimension 2^n to represent a quantum state defined by n qubits.

8.1.1 About the routine

- Qubitization is based on the block encoding technique.

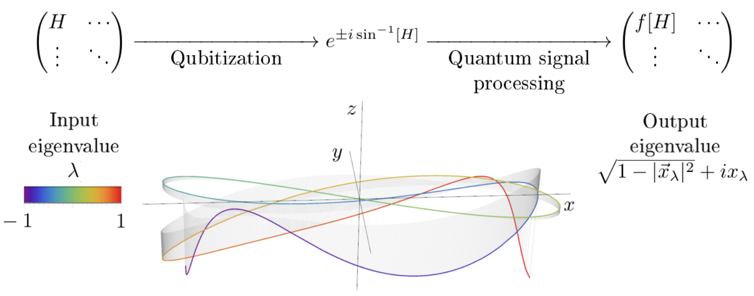


- It relies on the decomposition of the 2^{n+1} dimensional space on which U acts into 2^n spaces, each of dimension 2.
- Each such space of dimension 2 is generated by an eigenvector of A and another associated vector.
- Thus, qubitization refers to this decomposition in spaces of dimension 2.
- Each 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.
- 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 $e^{i\lambda t}$.
- This above 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.


$\begin{pmatrix} H & \cdots \\ \vdots & \ddots \end{pmatrix}$
Qubitization
$e^{\pm i\sin^{-1}[H]}$
Quantum signal processing
$\begin{pmatrix} f[H] & \cdots \\ \vdots & \ddots \end{pmatrix}$
Input eigenvalue λ
-1
1
z
y
x
Output eigenvalue $\sqrt{1-|\vec{x}_\lambda|^2}+ix_\lambda$

8.2 Wow qubitization rocks! Right? ... umm it does right?

Now that we have a vague idea of how to go about implementing qubitization, it is worth noting that we require a few (2) subroutines (namely SELECT and PREPARE) to serve as interfaces for the Hamiltonian.

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. This same problems persists, to some extent, in the case of "Truncated Taylor Series" method and "Szegedy Quantum Walk", as well.

However that said, qubitization is way better in simulating chemistry problems where there do not exist any self evident symmetries that can be taken advantage of.

8.3 Implementation

Let $H = \sum_j h_j H_j$ for unitary and Hermitian H_j and $h_j \geqslant 0$. Thus, we need to implement $W = e^{\pm i \cos^{-1}(H/|h|_1)}$ where $|h|_1 = \sum_j |h_j|$.

Now, consider the following subroutines:

- **Reflection**: We have SELECT $|j\rangle |\psi\rangle = |j\rangle H_j |\psi\rangle$. Now, SELECT^2 $|j\rangle |\psi\rangle = |j\rangle |\psi\rangle$ and the operator has eigenvalues ± 1.
- Purpose of SELECT is to be able to access each of the H_js.
- Now, we shall define the operator PREPARE to access the coefficients h_js.

$$\text{PREPARE}\,|0\rangle = \sum_j \sqrt{\frac{h_j}{|h|_1}}\,|j\rangle$$

- Let us define $\mathcal{M}$ as follows.

$$\mathcal{M}\,|0\rangle^{\otimes n} = \begin{cases} -|x\rangle & \text{if } x = 0 \\ |x\rangle & \text{otherwise} \end{cases}$$

- Thus, $\mathcal{M}$ is a multiply controlled phase gate.
- **Another reflection**: $R = 1 - 2\,\text{PREPARE}\,|0\rangle\langle 0|\,\text{PREPARE}^\dagger$.
- Thus, we have $R = \text{PREPARE}(\mathcal{M})\,\text{PREPARE}^\dagger$

Thus, the walk operatoor can be implemented as $W = (\text{SELECT})\ (R)$ which again can be seen to implement an operator that is equivalent (up to an isometry) to $e^{\pm i \cos^{-1}(H/|h|_1)}$.

REFERENCES

[1] M. Szegedy. quantum speed-up of markov chain based algorithms. In *45th Annual IEEE Symposium on Foundations of Computer Science*.

[2] Richard Feynman. Simulating physics with computers.

[3] Wikipedia contributors. Quantum walk — Wikipedia, the free encyclopedia, 2022. [Online; accessed 28-April-2022].

[4] Salvador Elías Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, jul 2012.

[5] Shantanav Chakraborty, Kyle Luh, and Jeremie Roland. How Fast Do Quantum Walks Mix? *Physical Review Letters*, 124(5), feb 2020.

[6] Andrew M. Childs. On the Relationship Between Continuous and Discrete Time Quantum Walk. *Communications in Mathematical Physics*, 294(2):581–603, oct 2009.

[7] Dominic W. Berry and Andrew M. Childs. Black-Box Hamiltonian Simulation and Unitary Implementation. *Quantum Info. Comput.*, 12(1–2):29–62, jan 2012.

[8] Laura Clinton, Johannes Bausch, and Toby Cubitt. Hamiltonian simulation algorithms for near-term quantum hardware. *Nature Communications*, 12(1), aug 2021.

[9] Yulong Dong, K. Birgitta Whaley, and Lin Lin. A Quantum Hamiltonian Simulation Benchmark, 2021.

[10] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019.

[11] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, oct 2015.

[12] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1), feb 2021.

[13] Jonathan Wei Zhong Lau, Tobias Haug, Leong Chuan Kwek, and Kishor Bharti. Nisq algorithm for hamiltonian simulation via truncated taylor series, 2021.

[14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. ACM, may 2014.

A QUANTUM WALKS

Theorem 1. *The quantum states defined by Childs in 6.1* $|\psi_1\rangle, \ldots, |\psi_N\rangle$ *are orthonormal.*

Proof.

$$\langle \psi_j | \psi_k \rangle = \frac{1}{||\text{abs}(H)||} \sum_{l_1, l_2 = 1}^{N} \sqrt{H_{jl_1} H^*_{kl_2} \frac{d_{l_1} d_{l_2}}{d_j d_k}} \langle j, l_1 | k, l_2 \rangle$$

Thus, it is clear that $\langle \psi_j | \psi_k \rangle = 0$ when $j \neq k$, that is the quantum states are orthogonal.

$$\begin{aligned}\langle \psi_j | \psi_j \rangle &= \frac{1}{||\text{abs}(H)||} \sum_{k_1, k_2 = 1}^{N} \sqrt{H_{jk_1} H^*_{jk_2} \frac{d_{k_1} d_{k_2}}{d_j^2}} \langle j, k_1 | j, k_2 \rangle \\ &= \frac{1}{||\text{abs}(H)||} \sum_{k=1}^{N} |H_{jk}| \frac{d_k}{d_j} \\ &= 1, \quad \text{since } \text{abs}(H) |d\rangle = ||\text{abs}(H)|| \, |d\rangle\end{aligned}$$

Hence, the quantum states are orthonormal. □

Theorem 2. *Suppose that for the Hamiltonian to be simulated* H*, with* abs(H) *defined in 6.1,* $\frac{H}{||abs(H)||} |\lambda\rangle = \lambda |\lambda\rangle$*. Then the szegedy walk operator* U *defined in 6.1.2 has two normalised eigenvectors*

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

with eigenvalues $\mu_\pm = \pm e^{\pm i \arcsin \lambda}$.

Proof. Considering the action of U on the vector $T|\lambda\rangle$, we have

$$\begin{aligned} UT|\lambda\rangle &= iS(2TT^\dagger - \mathbb{I})T|\lambda\rangle \\ &= 2iSTT^\dagger T|\lambda\rangle - iST|\lambda\rangle \\ &= iST|\lambda\rangle & \text{(since } T^\dagger T = \mathbb{I}) \end{aligned}$$

Considering the effect of the swap operator on the inner product of two of our orthonormal states we get

$$\begin{aligned}\langle \psi_j | S | \psi_k | \psi_j | S | \psi_k \rangle &= \frac{1}{||\text{abs}(H)||} \sum_{l_1, l_2 = 1}^{N} \sqrt{H_{jl_1} H^*_{kl_2} \frac{d_{l_1} d_{l_2}}{d_j d_k}} \langle j, l_1 | l_2, k \rangle \\ &= \frac{H_{jk}}{||\text{abs}(H)||}\end{aligned}$$

Thus, we also get the following.

$$\begin{aligned} T^\dagger S T &= \sum_{j,k=1}^{N} |j\rangle \langle \psi_j | S | \psi_k \rangle \langle k| \\ &= \frac{H}{||\text{abs}(H)||} \end{aligned}$$

And now, we have:

$$\begin{aligned} UST|\lambda\rangle &= iS(2TT^\dagger - \mathbb{I})ST|\lambda\rangle \\ &= 2iSTT^\dagger ST|\lambda\rangle - iS^2 T|\lambda\rangle \\ &= 2\lambda iST|\lambda\rangle - iT|\lambda\rangle \end{aligned}$$

The unnormalised eigenvector, let it be $|\mu\rangle$, is

$$\begin{aligned}|\mu\rangle &= \left(\mathbb{I} - e^{i\arccos\lambda}S\right) T|\lambda\rangle \\ &= \left(\mathbb{I} - ie^{i\arcsin\lambda}S\right) T|\lambda\rangle \\ &= T|\lambda\rangle + i\mu ST|\lambda\rangle\end{aligned}$$

Thus, we can now obtain an expression by equating $U|\mu\rangle = \mu|\mu\rangle$

$$\begin{aligned}U|\mu\rangle = \mu|\mu\rangle &\iff UT|\lambda\rangle + i\mu UST|\lambda\rangle = \mu T|\lambda\rangle + i\mu^2 ST|\lambda\rangle \\ &\iff iST|\lambda\rangle + i\mu\left(2\lambda iST|\lambda\rangle - iT|\lambda\rangle\right) = \mu T|\lambda\rangle + i\mu^2 ST|\lambda\rangle \\ &\iff i(1+2i\lambda\mu)ST|\lambda\rangle = i\mu^2 ST|\lambda\rangle \\ &\iff \mu^2 = 1 + 2i\lambda\mu\end{aligned}$$

Thus, solving the quadratic equation obtained, we obtain the unnormalised eigenvalues

$$\mu = i\lambda \pm \sqrt{1-\lambda^2} = ie^{\arccos\lambda} = \pm e^{\arcsin\lambda}$$

And the normalisation factor is

$$\begin{aligned}\langle\mu|\mu\rangle &= \left\langle\lambda|T^\dagger T|\lambda\right\rangle + i\mu\left\langle\lambda|T^\dagger ST|\lambda\right\rangle - i\mu^*\left\langle\lambda|T^\dagger S^\dagger T|\lambda\right\rangle + |\mu|^2\left\langle\lambda|T^\dagger S^\dagger ST|\lambda\right\rangle \\ &= 1 + i\mu\lambda - i\mu^*\lambda + |\mu|^2 \\ &= 2(1-\lambda^2), \quad \text{since } \mu = i\lambda \pm \sqrt{1-\lambda^2}\end{aligned}$$

Hence, by combining these results we get our the expressions for eigenvalues and eigenvectors that we wanted to prove. □

B HAMILTONIAN SIMULATION USING QUANTUM WALK

Theorem 3. *For the isometry mapping* T *defined in 6.1.2, and for an eigenvector* λ *of* $\frac{H}{||abs(H)||}$

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

where $\mu_\pm$ *and* $|\mu_\pm\rangle$ *are the eigenvalues and eigenvectors of the szegedy walk operator (ref 6.1.2)*

Proof. Let $x, y \in \mathbb{C}$ such that

$$T|\lambda\rangle = x|\mu_+\rangle + y|\mu_-\rangle$$

Then,

$$\begin{aligned}T|\lambda\rangle &= x|\mu_+\rangle + y|\mu_-\rangle \\ &= \frac{x+y}{\sqrt{2(1-\lambda^2)}}T|\lambda\rangle - \frac{xe^{i\arccos\lambda} + ye^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}ST|\lambda\rangle\end{aligned}$$

Thus, we get a system of linear equations as follows.

$$x + y = \sqrt{2(1-\lambda^2)}, xe^{i\arccos\lambda} + ye^{-i\arccos\lambda} = 0$$

And we solve this system to get the coefficients to be

$$x = \frac{1-\lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}, y = \frac{1-\lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}$$

and from that, we finally get

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

□

Theorem 4. *For the isometric mappings* T *and* P *defined in 6.1.2 and 7 respectively and the unitary operation defined in* **??***def*

$$\left\langle \psi|e^{iHt}T^{\dagger}P^{\dagger}F_tPT|\psi\right\rangle = \sum_{\lambda,\phi}|\psi_\lambda|^2 e^{i(\lambda-\sin\phi)t}\frac{\left|a_{\phi|\arcsin\lambda}\right|^2+\left|a_{\phi|\pi-\arcsin\lambda}\right|^2}{2}$$

Proof. First, we derive

$$\begin{aligned}
PTe^{-iHt}\left|\lambda\right\rangle &= PT\left(\sum_{\lambda'}e^{-i\lambda' t}\left|\lambda'\right\rangle\left\langle\lambda'\right|\right)\left|\lambda\right\rangle \\
&= e^{-i\lambda t}PT\left|\lambda\right\rangle \\
&= e^{-i\lambda t}P\left(\frac{1-\lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_+\right\rangle + \frac{1-\lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_-\right\rangle\right) \\
&= e^{-i\lambda t}\sum_{\phi}\Bigg(a_{\phi|\arcsin\lambda}\frac{1-\lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_+,\phi\right\rangle + \\
&\qquad a_{\phi|\pi-\arcsin\lambda}\frac{1-\lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_-,\phi\right\rangle\Bigg)
\end{aligned}$$

And thus,

$$\begin{aligned}
F_tPT\left|\lambda\right\rangle = \sum_{\phi}e^{-it\sin\phi}\Bigg(&a_{\phi|\arcsin\lambda}\frac{1-\lambda e^{-i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_+,\phi\right\rangle + \\
&a_{\phi|\pi-\arcsin\lambda}\frac{1-\lambda e^{i\arccos\lambda}}{\sqrt{2(1-\lambda^2)}}\left|\mu_-,\phi\right\rangle\Bigg)
\end{aligned}$$

Using orthonormality of eigenstates (1), and the previous observation, we derive the required inner product to be

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

□

Theorem 5. *For the expression derived for inner product between the expected and predicted states of simulating the Hamiltonian* H *for time* t *in 4, derive that the fidelity of the simulation is lower bounded by*

$$1-\frac{t^2}{2}\min_{\theta\in[0,2\pi)}\sum_{\phi}(\theta-\phi)^2|a_{\phi|\theta}|^2$$

Proof. Fidelity of the simulation is $\left|\left\langle\psi|e^{iHt}T^{\dagger}P^{\dagger}F_tPT|\psi\right\rangle\right|$ and the first step at taking the lower bound is to minimise the value on one parameter of the summation.

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

and this holds because, $|\sin\theta-\sin\phi|\leqslant|\theta-\phi|$. □

Theorem 6. *With the assumptions made in 7, the lower bound for the fidelity of the simulation is* $1-93t^2/M^2$*.*

Proof. With the given assumptions, for sufficiently large M,

$$\begin{aligned}
|a_{\theta+\Delta|\theta}|^2 &\leqslant \frac{\pi^2}{2M^4\sin^2\left(\frac{\Delta}{2}+\frac{\pi}{2(M+1)}\right)\sin^2\left(\frac{\Delta}{2}-\frac{\pi}{2(M+1)}\right)}\\
&\leqslant \frac{128\pi^2}{M^4\Delta^4\left(1-\frac{\pi^2}{\Delta^2(M+1)^2}\right)^2}\\
&\leqslant \frac{512\pi^2}{9M^4\Delta^4}
\end{aligned}$$

Here, the last step assumes that $\Delta\geqslant 2\pi/M$. Now, since we have $2\pi j/M\leqslant\Delta\leqslant 2\pi(j+1)/M$, the following holds.

$$\begin{aligned}
\sum_{\phi}(\theta-\phi)^2|a_{\phi|\theta}|^2 &\leqslant \frac{4\pi^2}{M^2}+2\sum_{j=1}^{\infty}\frac{128(j+1)^2}{9M^2j^2}\\
&= \frac{4\pi^2}{M^2}+\frac{256}{9M^2}\left(\frac{15\pi^2+\pi^4+180\zeta(3)}{90}\right)\\
&\leqslant \frac{186}{M^2}
\end{aligned}$$

Now considering the previously derived bound on fidelity of the simulation, we get the value to be

$$\begin{aligned}
\left|\left\langle\psi|e^{iHt}T^{\dagger}P^{\dagger}F_tPT|\psi\right\rangle\right| &\geqslant 1-\frac{t^2}{2}\min_{\theta\in[0,2\pi)}\sum_{\phi}(\theta-\phi)^2|a_{\phi|\theta}|^2\\
&\geqslant 1-\frac{t^2}{2}\cdot\frac{186}{M^2}=1-\frac{93t^2}{M^2}
\end{aligned}$$

□

C DECOMPOSITION LEMMA

Theorem 7. *Let* H *be a* d*-sparse, row-computable Hamiltonian over* n *qubits. It is possible to decompose* H *into* $H=\sum_{m=1}^{(d+1)^2n^6}H_m$ *where each* H_m *is:*

- *A sparse, row-computable Hamiltonian over n qubits, and,*
- *A 2×2 combinatorially block diagonal.*

Proof. For the Hamiltonian H, we label each entry of the matrix $H_{i,j}$ for $i < j$ (upper diagonal entries) by a colour ^{1}. The colour of an entry $\mathrm{col}_H(i,j)$ is defined as the tuple $(k, i \bmod k, j \bmod k, r_H(i,j), c_H(i,j))$ where

$$k = \begin{cases} 1 & \text{if } i = j \\ \text{The lowest integer where } i \neq j \text{ mod } \text{ for } 2 \leqslant k \leqslant n^2 & \text{otherwise} \end{cases}$$

and

$$r_H(i,j) = \begin{cases} 0 & \text{if } H_{i,j} = 0 \\ \text{The index of } H_{i,j} \text{ in the list of all} & \\ \text{non-zero elements in the } i^{th} \text{ row of } H & \text{otherwise.} \end{cases}$$

The definition of $c_H(i,j)$ is the same as Equation (16) but now for columns as opposed to rows. For values of $i > j$ (lower diagonal entries), we define $\mathrm{col}_H(i,j) = \mathrm{col}_H(j,i)$. We define H_m to contain all entries from H that are a particular colour m. As each entry of H_m has a single assigned colour, we can reconstruct H by a summation over all colours ($H = \sum H_m$). As H is Hermitian and each H_m is symmetric about the diagonal (from $\mathrm{col}_H(i,j) = \mathrm{col}_H(j,i)$), it follows that every H_m is also Hermitian. Also as H is row sparse and row computable, it follows that each H_m also has these properties. From this, one can see that the first requirement of the theorem is satisfied.

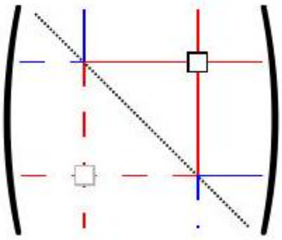


Figure 3: How the colouring of the matrix restricts the possible structure of H_m

From the restraint of $r_H(i,j)$ and $c_H(i,j)$ on the colour tuple, we know that for a non-zero element of a matrix H_m at position (i,j), there are no other non-zero elements in row i or column j (red lines in upper half of Figure 1). Furthermore, the other tuple colouring components $(k, i \bmod k, j \bmod k)$ create the case where there exists only a single non-zero element of H_m in the row j or column i in the position (j,i) (blue lines in upper half of Figure 1). This case is reflected for $i < j$ as $\mathrm{col}_H(i,j) = \mathrm{col}_H(j,i)$ (dashed lines in Figure 1). From these two conditions, it leads that each matrix H_m can be permuted to become a 2×2 block diagonal matrix where each mirror diagonal pair $(H_{i,j}, H_{j,i})$ forms a block. Therefore each H_m is a combinatorially block diagonal, proving the second part of the theorem. □