Limitations on the simulation of non-sparse Hamiltonians

Andrew M. Childs[*]

Department of Combinatorics & Optimization
and Institute for Quantum Computing
University of Waterloo

Robin Kothari[†]

David R. Cheriton School of Computer Science
and Institute for Quantum Computing
University of Waterloo

Abstract

The problem of simulating sparse Hamiltonians on quantum computers is well studied. The evolution of a sparse $N \times N$ Hamiltonian H for time t can be simulated using $O(\|Ht\| \operatorname{poly}(\log N))$ operations, which is essentially optimal due to a no–fast-forwarding theorem. Here, we consider non-sparse Hamiltonians and show significant limitations on their simulation. We generalize the no–fast-forwarding theorem to dense Hamiltonians, ruling out generic simulations taking time $o(\|Ht\|)$, even though $\|H\|$ is not a unique measure of the size of a dense Hamiltonian H. We also present a stronger limitation ruling out the possibility of generic simulations taking time $\operatorname{poly}(\|Ht\|, \log N)$, showing that known simulations based on discrete-time quantum walk cannot be dramatically improved in general. On the positive side, we show that some non-sparse Hamiltonians can be simulated efficiently, such as those with graphs of small arboricity.

1 Introduction

One of the primary applications of quantum computers is the simulation of quantum systems. Indeed, it was the apparent exponential time complexity of simulating quantum systems on a classical computer that led Feynman to propose the idea of quantum computation [14].

In addition to predicting the behavior of physical systems, Hamiltonian simulation has algorithmic applications. For example, the implementation of a continuous-time quantum walk algorithm is a Hamiltonian simulation problem. Examples of algorithms that can be implemented using Hamiltonian simulation methods include unstructured search [13], adiabatic optimization [12], a quantum walk with exponential speedup over classical computation [8], and the recent NAND tree evaluation algorithm [10].

In the Hamiltonian simulation problem, our goal is to implement the unitary operator e^{-iHt} for some given Hamiltonian H and time t. We say that a Hamiltonian H acting on an N-dimensional quantum system can be simulated efficiently if there is a quantum circuit using $\operatorname{poly}(\log N, t, 1/\epsilon)$ one- and two-qubit gates that approximates (with error at most ϵ) the evolution according to H for time t. (Of course, we can rescale t by rescaling H, so the complexity of simulating H for time t must also depend on some measure of the size of H, as discussed in more detail below.)

Efficient simulations are known for various classes of Hamiltonians. For example, a Hamiltonian for a system of qubits can be simulated efficiently whenever it is *local*, meaning that it is a sum of terms, each of which acts on a constant number of qubits [18]. More generally, a Hamiltonian H can be simulated efficiently if it is *sparse* (i.e., has only $\operatorname{poly}(\log N)$ nonzero entries per row) and

[*]amchilds@uwaterloo.ca

[†]rkothari@cs.uwaterloo.ca

efficiently row-computable (i.e., there is an efficient means of computing the indices and the matrix elements of the nonzero entries in any given row) [1].

These conditions lead to a convenient black-box formulation of the problem, in which a black box can be queried with a row index j and an index i to obtain the ith nonzero entry in the jth row. This black box can be implemented efficiently provided that H is efficiently row-computable. A series of results has decreased the number of black-box queries, in terms of N, from the original $O(\log^9 N)$ [1], to $O(\log^2 N)$ [6], to $O(\log^* N)$ [4]. In particular, Berry, Ahokas, Cleve, and Sanders [4] present an almost linear-time algorithm for simulating sparse Hamiltonians with query complexity

$$(\log^* N)d^4\|Ht\|\left(\frac{\|d^2Ht\|}{\epsilon}\right)^{o(1)}, \tag{1}$$

where d is the maximum number of nonzero entries in any row and ϵ is the maximum error permitted in the final state (quantified in terms of trace distance).

The dependence of (1) on the simulation time is nearly optimal, since it is not possible to simulate a general sparse Hamiltonian for time t using $o(t)$ queries. Intuitively, there is no generic way to fast-forward through the time evolution of quantum systems. More formally,

Theorem 1 (No–fast-forwarding theorem [4, Theorem 3])**.** *For any positive integer N there exists a row-computable sparse Hamiltonian H with $\|H\| = 1$ such that simulating the evolution of H for time $t = \pi N/2$ within precision $1/4$ requires at least $N/4$ queries to H.*

More recently, methods have been presented for simulating a Hamiltonian H that is not necessarily sparse. Of course, we do not expect to efficiently simulate a general Hamiltonian, simply because there are too many Hamiltonians to consider (just as we cannot hope to efficiently implement a general unitary operation [17]). However, we can conceivably efficiently simulate non-sparse Hamiltonians with a suitable concise description. In particular, by applying phase estimation to a discrete-time quantum walk derived from H, one can simulate H for time t in a number of walk steps that grows only linearly with t [7]. More precisely, we have

Theorem 2. *For any Hermitian matrix H, there is a discrete-time quantum walk on the graph of nonzero entries of H such that $e^{-\mathrm{i}Ht}$ can be simulated with error at most δ using $O(\|\mathrm{abs}(Ht)\|/\sqrt{\delta})$ steps of the walk, where $\mathrm{abs}(H)$ is the matrix with entries $\mathrm{abs}(H)_{jk} = |H_{jk}|$.*

Of course, to apply this result, we must implement the discrete-time quantum walk derived from H. This can be done efficiently for various concisely specified non-sparse Hamiltonians [7]. Note that the same theorem holds with $\|\mathrm{abs}(H)\|$ replaced by $\|H\|_1$ (a matrix norm defined in Section 2); this quantity is generally larger than $\|\mathrm{abs}(H)\|$, but the resulting walk may be easier to implement.

Notice that the overhead of this simulation is proportional not to the spectral norm $\|H\|$, but to a measure of the size of H that can be much larger when some entries of H are negative (or more generally, complex). This naturally raises the question of whether an improved simulation is possible. In the present article, we examine this possibility. Unfortunately, our main result is negative: there is no general Hamiltonian simulation algorithm that uses only $\mathrm{poly}(\|Ht\|, \log N)$ steps (Theorem 4).

The remainder of this article is organized as follows. In Section 2, we introduce various matrix norms that arise when quantifying the complexity of Hamiltonian simulation and relate them to one another. We then move on to lower bounds for non-sparse Hamiltonians in Section 3, where we describe how the no–fast-forwarding theorem can be modified to give a lower bound that depends on the spectral norm rather than various smaller measures of the size of a Hamiltonian. Then, in

Section 4, we present the main result, an example of a family of Hamiltonians with $\|\mathrm{abs}(H)\| \gg \|H\|$ that cannot be simulated in time $\mathrm{poly}(\|Ht\|, \log N)$. We then turn to upper bounds in Section 5, and investigate how certain structured Hamiltonians can be simulated in time $O(\|Ht\|)$—in particular, we give a positive result on the simulation of Hamiltonians whose graphs have small arboricity.[1] Finally, we conclude in Section 6 with a discussion of open problems.

2   Measures of simulation complexity

Upper and lower bounds on the complexity of simulating a Hamiltonian H depend on some measure of the size of H. Since $e^{-\mathrm{i}Ht}$ depends only on the product Ht, the complexity of simulating H for time t is some function of Ht. For example, the no–fast-forwarding theorem clearly cannot be circumvented by simply multiplying H by a constant. Similarly, simulation results such as those for sparse Hamiltonians, using $\|Ht\|^{1+o(1)}$ operations, and Theorem 2, using $O(\|\mathrm{abs}(Ht)\|)$ operations, depend on various measures of the size of Ht.

In this section, we take a step back and consider properties of various measures of the size of H that may play a role in the complexity of simulating it. Let $\nu(Ht)$ be a function that measures the complexity of simulating H for time t. We can infer various properties of $\nu(\cdot)$ as follows. Since it is trivial to simulate the identity operation, $\nu(0) = 0$. On the other hand, if $H \neq 0$, then it requires some work to simulate, so $\nu(H) > 0$. It is also plausible to suppose that $\nu(tH) = |t|\nu(H)$. We clearly have $\nu(Ht) \leq |t|\nu(H)$ for $t \in \mathbb{Z}$, since Ht can be simulated using t exact simulations of H. On the other hand, the no–fast-forwarding theorem suggests that this is the best possible way to simulate Ht in general. Finally, since the Lie product formula can be used to simulate $H + K$ using simulations of H and K, we expect that $\nu(H + K) \lessapprox \nu(H) + \nu(K)$ (up to the fact that a bounded-error simulation requires a slightly superlinear number of operations).

These properties are reminiscent of the axioms for matrix norms, suggesting that it may be reasonable to quantify the complexity of simulating H in terms of some matrix norm $\nu(H)$. Indeed, results on the simulation of sparse Hamiltonians are typically stated in terms of the spectral norm $\|H\|$, and Theorem 2 also involves matrix norms. We now introduce various matrix norms relevant to Hamiltonian simulation.

Definition 1 (Spectral norm)**.** The spectral norm of a matrix H is defined as

$$\|H\| := \max_{v \neq 0} \frac{\|Hv\|}{\|v\|} = \max_{\|v\|=1} \|Hv\|, \tag{2}$$

where $\|v\|$ is the standard Euclidean vector norm defined as $\|v\| := \sqrt{\sum_i |v_i|^2}$.

The spectral norm, also known as the operator norm or induced Euclidean norm, is equal to the largest singular value of the matrix. For Hermitian matrices it is also equal to the magnitude of the largest eigenvalue. This norm arises in the complexity of sparse Hamiltonian simulation algorithms, and in Theorem 2 as the spectral norm of $\mathrm{abs}(H)$, the matrix with entries $\mathrm{abs}(H)_{jk} = |H_{jk}|$.

Definition 2 (Induced 1-norm)**.** The induced 1-norm of a matrix H is defined as

$$\|H\|_1 := \max_{v \neq 0} \frac{\|Hv\|_1}{\|v\|_1} = \max_j \sum_i |H_{ij}|, \tag{3}$$

where $\|v\|_1$ is the vector 1-norm defined as $\|v\|_1 := \sum_i |v_i|$.

---

[1]A graph is said to have *arboricity* k if its adjacency matrix can be written as the sum the adjacency matrices of k forests, but not $k - 1$ forests.

The induced 1-norm is equal to the maximum absolute column sum of the matrix. As mentioned in Section 1, Theorem 2 holds with $\|\mathrm{abs}(H)\|$ replaced by $\|H\|_1$. This does not, however, lead to a superior simulation method since $\|H\|_1 \geq \|\mathrm{abs}(H)\|$, as shown in Lemma 1 below.

Definition 3 (Maximum column norm). The maximum column norm of a matrix H is defined as

$$\mathrm{mcn}(H) := \max_j \sqrt{\sum_i |H_{ij}|^2} = \max_{v \neq 0} \frac{\|Hv\|}{\|v\|_1} = \max_j \|He_j\|, \tag{4}$$

where e_j is the jth column of the identity matrix.

The maximum column norm is the maximum Euclidean norm of the columns of H. This norm appears in the complexity of an algorithm for simulating Hamiltonians whose graphs are trees [7, Theorem 4] and in the related Proposition 2 in Section 5.

Definition 4 (Max norm). The max norm of a matrix H is defined as

$$\max(H) := \max_{i,j} |H_{ij}|. \tag{5}$$

The max norm is just the largest entry of H in absolute value. It is a matrix norm, and is typically much smaller than the other norms mentioned.

The following lemma relates the various norms introduced above.

Lemma 1. *For any Hermitian matrix $H \in \mathbb{C}^{N \times N}$, we have the following inequalities:*

$$\max(H) \leq \mathrm{mcn}(H) \leq \|H\| \leq \|\mathrm{abs}(H)\| \leq \|H\|_1 \leq \sqrt{N}\,\mathrm{mcn}(H) \leq N \max(H). \tag{6}$$

Furthermore, each of these inequalities is the best possible.

Proof. The first inequality follows from the fact that the maximum element in any column cannot be greater than the Euclidean norm of that column. We have

$$\max(H) = \max_j \left(\max_i |H_{ij}| \right) \leq \max_j \sqrt{\sum_i |H_{ij}|^2} = \mathrm{mcn}(H). \tag{7}$$

The next inequality follows from the observation that $\mathrm{mcn}(H)$ is defined by a maximum over the standard basis vectors e_j, whereas $\|H\|$ is defined by a maximum over all vectors with norm 1, which contains the set of all e_j. Thus

$$\mathrm{mcn}(H) = \max_j \|He_j\| \leq \max_{\|v\|=1} \|Hv\| = \|H\|. \tag{8}$$

Using the triangle inequality with $\|H\| = \max_{\|v\|=1} (\sum_i |\sum_j H_{ij}v_j|^2)^{\frac{1}{2}}$, we get

$$\|H\| \leq \max_{\|v\|=1} \left(\sum_i \left| \sum_j |H_{ij}||v_j| \right|^2 \right)^{\frac{1}{2}} = \max_{\substack{\|v\|=1 \\ v_j \geq 0}} \left(\sum_i \left| \sum_j \mathrm{abs}(H)_{ij}v_j \right|^2 \right)^{\frac{1}{2}}. \tag{9}$$

Now by maximizing over all v with $\|v\| = 1$ instead of only those with $v_j \geq 0$, we get

$$\max_{\substack{\|v\|=1 \\ v_j \geq 0}} \left(\sum_i \left| \sum_j \mathrm{abs}(H)_{ij}v_j \right|^2 \right)^{\frac{1}{2}} \leq \max_{\|v\|=1} \left(\sum_i \left| \sum_j \mathrm{abs}(H)_{ij}v_j \right|^2 \right)^{\frac{1}{2}} = \|\mathrm{abs}(H)\|. \tag{10}$$

4

The last inequality is actually an equality due to the Perron–Frobenius theorem.

Since $\mathrm{abs}(H)$ is a symmetric matrix, there is an eigenvector z with eigenvalue equal in magnitude to $\|\mathrm{abs}(H)\|$. Clearly this eigenvector satisfies $\|\mathrm{abs}(H)z\|_1 = \|\mathrm{abs}(H)\|\|z\|_1$. Using this and maximizing over all nonzero vectors, we have

$$\|\mathrm{abs}(H)\| = \frac{\|\mathrm{abs}(H)\|\|z\|_1}{\|z\|_1} = \frac{\|\mathrm{abs}(H)z\|_1}{\|z\|_1} \leq \max_{v \neq 0} \frac{\|\mathrm{abs}(H)v\|_1}{\|v\|_1} = \|\mathrm{abs}(H)\|_1. \tag{11}$$

The inequality now follows from the fact that $\|H\|_1 = \|\mathrm{abs}(H)\|_1$, since

$$\|\mathrm{abs}(H)\|_1 = \max_j \sum_i |\mathrm{abs}(H)_{ij}| = \max_j \sum_i |H_{ij}| = \|H\|_1. \tag{12}$$

For the next inequality, we use the fact that $\|v\|_1 \leq \sqrt{N}\|v\|$ for all vectors v. This can be proved using the Cauchy–Schwarz inequality, $|\langle u, v \rangle| \leq \|u\|\|v\|$, by taking $u_i = v_i/|v_i|$. Let $j_{\max}$ be the index j that maximizes $\sum_i |H_{ij}|$. Thus $\|H\|_1 = \sum_i |H_{ij_{\max}}| = \|He_{j_{\max}}\|_1$. Using these two inequalities, it follows that

$$\|H\|_1 = \|He_{j_{\max}}\|_1 \leq \sqrt{N}\|He_{j_{\max}}\| \leq \sqrt{N} \max_j \|He_j\| = \sqrt{N}\,\mathrm{mcn}(H). \tag{13}$$

The last inequality is proved using the fact that for any j, $H_{ij} \leq \max_i H_{ij}$; thus

$$\mathrm{mcn}(H) = \max_j \sqrt{\sum_i |H_{ij}|^2} \leq \max_j \sqrt{N \max_i |H_{ij}|^2} = \sqrt{N} \max_{ij} |H_{ij}| = \sqrt{N}\max(H). \tag{14}$$

For each of these inequalities, there is a matrix that achieves equality. The first four inequalities are saturated when H is the identity matrix since the relevant norms are all equal to 1. The last two inequalities are satisfied with equality when H is the all-ones matrix (i.e., for all i, j, $H_{ij} = 1$), since then $\|H\|_1 = N$, $\mathrm{mcn}(H) = \sqrt{N}$, and $\max(H) = 1$. ∎

Since Theorem 2 involves $\|\mathrm{abs}(H)\|$, we would like to relate $\|\mathrm{abs}(H)\|$ and $\|H\|$. Lemma 1 gives $\|\mathrm{abs}(H)\| \leq \sqrt{N}\|H\|$, which is also the best possible inequality between the two norms. For example, when N is a power of 2, the matrix $H = R^{\otimes \log N}$ achieves equality, where $R := \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix} \right)/\sqrt{2}$ is the Hadamard matrix. It has $\|H\| = 1$, but $\|\mathrm{abs}(H)\| = \sqrt{N}$. This shows that Theorem 2 might not be as powerful as we would like, since for some Hamiltonians, the simulation method of Theorem 2 may be infeasible even when $\|H\|$ is small.

Although the above inequalities cannot be tightened in general, there can of course be stronger relationships among the various norms for special classes of Hamiltonians. For example, if H is sparse, observe that the norms mentioned above can differ at most by a factor of $\mathrm{poly}(\log N)$. Specifically, if H is k-sparse (i.e., it has at most k nonzero entries per row), then

$$\max(H) \leq \mathrm{mcn}(H) \leq \|H\| \leq \|\mathrm{abs}(H)\| \leq \|H\|_1 \leq \sqrt{k}\,\mathrm{mcn}(H) \leq k\max(H). \tag{15}$$

The first four inequalities are from Lemma 1. The inequality $\|H\|_1 \leq \sqrt{k}\,\mathrm{mcn}(H)$ follows from (13) using the fact that $\|v\|_1 \leq \sqrt{k}\|v\|$ when v has at most k non-zero entries (this can be proved using the Cauchy–Schwarz inequality as before). The last inequality follows from (14) and the inequality $\sum_i |H_{ij}|^2 \leq k \max_i |H_{ij}|^2$, which holds when H is k-sparse. Thus the choice of norm for sparse matrices is quite flexible, since all the above-mentioned norms are equivalent up to polynomial factors.

In Section 5, we discuss some more examples in which Lemma 1 can be strengthened, with emphasis on the implications for simulations.

3   A no–fast-forwarding theorem for dense Hamiltonians

The no–fast-forwarding theorem (Theorem 1 above) establishes a lower bound for the simulation of sparse Hamiltonians. Although we stated the theorem with $\|H\| = 1$, any of the norms in Lemma 1 could have been used, since the Hamiltonian used in the proof of the no–fast-forwarding theorem is 2-sparse, and by (15) the norms differ at most by a factor of 2. In particular, the theorem could be restated with $\max(H) \leq 1$ or $\|H\|_1 \leq 2$.

Since the choice of norm is unclear, it is conceivable that there are Hamiltonian simulation algorithms that run in time $O(\max(Ht))$ or $O(\mathrm{mcn}(Ht))$. To distinguish between the norms, we require a dense Hamiltonian. The aim of this section is use the proof techniques of Theorem 1 to establish a similar theorem for dense Hamiltonians. In particular, we show that there does not exist an algorithm for simulating dense Hamiltonians in time $O(\max(Ht))$ or $O(\mathrm{mcn}(Ht))$. However, this does not appear to rule out $\mathrm{poly}(\|Ht\|)$ simulations, which we rule out in the next section. Although Theorem 4 in the next section is stronger than Theorem 3 below, we briefly present this straightforward generalization of the no–fast-forwarding theorem to show the extent of that approach as applied to the non-sparse case.

As in Theorem 1, we consider a black-box formulation of the problem of simulating dense Hamiltonians. There is a black box that can be queried with a row index j, which outputs the entire j^{th} row. Since the Hamiltonian is dense, this output can be exponentially large. This is not a problem, however, since our goal is to find a lower bound on query complexity, not time complexity. Even though each query takes exponential space, it counts as only one query. The black box used here is more powerful than the one in Theorem 1, so the lower bound proved below also carries over to the black box used in Theorem 1.

In terms of this black-box model, we have the following:

Theorem 3. *For any positive integer N, there exists a non-sparse Hamiltonian H such that simulating the evolution of H for time $t = \pi N/2$ within precision $1/4$ requires at least $N/4$ queries to H. This Hamiltonian has $\|H\| = 1$, $\mathrm{mcn}(H) = \Theta(1/\sqrt{N})$, and $\max(H) = \Theta(1/N)$.*

Proof. The main idea, as in the proof of Theorem 1 [4], is to construct a Hamiltonian whose simulation for time $t = \pi N/2$ determines the parity of N bits. Since we know that computing the parity of N bits requires at least $N/2$ queries [3, 11], this Hamiltonian cannot be simulated with $o(N)$ queries. Moreover, we want this Hamiltonian to be non-sparse.

We start with a simple Hamiltonian H_1 whose graph is just a line with $N + 1$ vertices. Consider the Hamiltonian acting on vectors $|i\rangle$ with $i \in \{0, \ldots, N\}$. The nonzero matrix entries of H_1 are $\langle i |H_1| i + 1 \rangle = \langle i + 1 |H_1| i \rangle = \sqrt{(N-i)(i+1)}/N$ for $i \in \{0, 1, \ldots, N-1\}$. This Hamiltonian has $\|H_1\| = 1$, and simulating H_1 for $t = \pi N/2$ starting with the state $|0\rangle$ gives the state $|N\rangle$ (i.e., $e^{-iH_1 t}|0\rangle = |N\rangle$).

Now, as in Ref. [4], consider a Hamiltonian H_2 generated from an N-bit string $S_0 S_1 \ldots S_{N-1}$. H acts on vertices $|i, j\rangle$, with $i \in \{0, \ldots, N\}$ and $j \in \{0, 1\}$. The nonzero matrix entries of this Hamiltonian are

$$\langle i, j |H_2| i + 1, j \oplus S_i \rangle = \langle i + 1, j \oplus S_i |H_2| i, j \rangle = \sqrt{(N-i)(i+1)}/N \qquad (16)$$

for all i and j. By construction, $|0, 0\rangle$ is connected to either $|i, 0\rangle$ or $|i, 1\rangle$ for any i; it is connected to $|i, j\rangle$ if and only if $j = S_0 \oplus S_1 \oplus \ldots \oplus S_{i-1}$. Thus $|0, 0\rangle$ is connected to either $|N, 0\rangle$ or $|N, 1\rangle$, and determining which is the case determines the parity of S. The graph of this Hamiltonian consists of two disjoint lines, one of which contains $|0, 0\rangle$ and either $|N, 0\rangle$ or $|N, 1\rangle$ depending on the parity of S. Just as for H_1, starting with the state $|0, 0\rangle$ and simulating H_2 for time $t = \pi N/2$ will give

6

either $|N, 0\rangle$ or $|N, 1\rangle$, which determines the parity of S. Note that since H_2 is a permutation of $H_1 \oplus H_1$, $\|H_2\| = \|H_1\| = 1$.

Finally, we construct the dense Hamiltonian H that has the properties stated in the theorem. As before, H is generated from an N-bit string $S_0 S_1 \dots S_{N-1}$. H acts on vertices $|i, j, k\rangle$, with $i \in \{0, \dots, N\}$, $j \in \{0, 1\}$, and $k \in \{0, N-1\}$. The nonzero entries of H are given by

$$\langle i, j, k \,|H|\, i+1, j \oplus S_i, k'\rangle = \langle i+1, j \oplus S_i, k' \,|H|\, i, j, k\rangle = \sqrt{(N-i)(i+1)}/N^2 \qquad (17)$$

for all i, j, k, and k'. The graph of H is similar to that of H_2, except that for each vertex in H_2, there are now N copies of it in H. This Hamiltonian is dense because it has $\Theta(N^2)$ vertices and each vertex is connected to all N copies of its neighboring vertices, which gives at least N nonzero entries in each row.

Now we simulate the Hamiltonian starting from the uniform superposition over the copies of the $|0, 0\rangle$ state, i.e., from the state $\frac{1}{\sqrt{N}} \sum_k |0, 0, k\rangle$. The subspace $\text{span}\{\sum_k |i, j, k\rangle\}$ of uniform superpositions over the third register is an invariant subspace of this Hamiltonian. Since the initial state lies in this subspace, the quantum walk remains in this subspace. In other words, the quantum walk on this dense graph starting from the chosen state reduces to the quantum walk on H_2 starting from the $|0, 0\rangle$ state.

Now, just as before, the parity of S can be determined by simulating H for time $t = \pi N/2$. This gives the lower bound of $N/2$ queries.

To calculate the norms of this Hamiltonian, we observe that $H = H_2 \otimes J/N$, where J is the all-ones matrix of size $N \times N$. This gives $\|H\| = \|H_2\| \cdot \|J\|/N = 1$. Direct computation shows that $\max(H) = \Theta(1/N)$ and $\text{mcn}(H) = \Theta(1/\sqrt{N})$. $\qquad \square$

This theorem rules out algorithms that make only $O(\max(Ht))$ or $O(\text{mcn}(Ht))$ queries, since for this Hamiltonian with $t = \pi N/2$ we have $\max(Ht) = \Theta(1)$ and $\text{mcn}(Ht) = \Theta(\sqrt{N})$, both of which are disallowed by the lower bound of $\Omega(N)$. However, this does not distinguish between $\|H\|$ and $\|\text{abs}(H)\|$ (since $\text{abs}(H) = H$), and in fact $\|H\|_1 \sim 1$ as well. In the next section we construct examples with $\|\text{abs}\, H\| \gg \|H\|$ in order to show that a general simulation using $O(\|Ht\|)$ steps (or even $\text{poly}(\|Ht\|, \log N)$ steps) is not possible.

4 A stronger limitation for dense Hamiltonians

As discussed in Section 1, there are dense Hamiltonian simulation algorithms that use $O(\|\text{abs}(Ht)\|)$ or $O(\|Ht\|_1)$ steps of a discrete-time quantum walk. However, in light of the no–fast-forwarding theorem for dense Hamiltonians, it might be reasonable to hope that dense Hamiltonians can be simulated in $O(\|Ht\|)$ steps, or at least in $\text{poly}(\|Ht\|, \log N)$ steps. Indeed, if such simulations existed they could be applied to give new quantum algorithms for various problems [7]. Unfortunately, in this section, we show that such simulations are not possible in general.

The currently known dense Hamiltonian simulation algorithms rely on certain properties of the Hamiltonian that we call its *structural properties*. By this we mean the location of nonzero entries in H, which correspond to the location of edges in the graph of the Hamiltonian, and the magnitudes of the edge weights. (The remaining information about the Hamiltonian is the phase of each matrix entry H_{ij}.)

Given the structural information, the currently known algorithms can simulate dense Hamiltonians using $O(\|\text{abs}(Ht)\|)$ or $O(\|Ht\|_1)$ calls to an oracle that gives the phase of the matrix entry at H_{ij}. We call this the *matrix entry phase oracle*. This oracle provides the value of $H_{ij}/|H_{ij}|$ when

queried with the input (i, j). (The oracle may return any complex number of unit modulus—say, 1—when $H_{ij} = 0$.)

We show that given this matrix entry phase oracle and complete structural information, there exist some Hamiltonians that cannot be simulated with $\mathrm{poly}(\|Ht\|, \log N)$ queries (although they can be simulated with $O(\|\mathrm{abs}(Ht)\|)$ queries [7]). The following theorem is our main result.

Theorem 4. *No quantum algorithm can simulate a general Hamiltonian $H \in \mathbb{C}^{N \times N}$ for time t with $\mathrm{poly}(\|Ht\|, \log N)$ queries to a matrix entry phase oracle, even when given complete structural information about the Hamiltonian.*

Proof. The proof of the theorem is divided into two parts. First we show that there exists a set of Hamiltonians of size $N \times N$ that is hard to simulate on average for a particular time t. Specifically, we show that simulating a Hamiltonian selected uniformly at random from this set for a chosen time has average-case query complexity $\Omega(\sqrt{N / \log N})$. Then we show that a Hamiltonian simulation algorithm that makes $\mathrm{poly}(\|Ht\|, \log N)$ queries would violate this lower bound. The lemmas used in the proof are proved in the appendix.

To show the lower bound, we need a black-box problem with an $\Omega(\sqrt{N / \log N})$ average-case lower bound, and a set of Hamiltonians whose simulation would solve this problem. We consider the problem of distinguishing strings $s \in \{-1, +1\}^M$ that have sum $-B$ or $+B$, given a black box for the entries of the string. When queried with an index $i \in \{1, 2, \ldots, M\}$, the black box returns the value of $s_i \in \{-1, +1\}$, where $s = s_1 s_2 \ldots s_M$. The following lemma characterizes the query complexity of this problem.

Lemma 2. *Suppose we are given black-box access to a string $s \in \{-1, +1\}^M$, where s is chosen uniformly at random from the set of strings with $\sum_i s_i \in \{-B, +B\}$. Then determining $\sum_i s_i$ has average-case quantum query complexity $\Theta(M/B)$.*

Thus, determining whether the sum is $-\sqrt{M \log M}$ or $+\sqrt{M \log M}$, with the promise that one of these is the case, requires $\Omega(\sqrt{M / \log M})$ quantum queries on average. For each string s, we construct a Hamiltonian H_s whose simulation for a particular time allows us to distinguish the two possible cases assuming s satisfies the promise.

Let H_s be a symmetric circulant matrix of size $N \times N$, where $N = 2M + 1$ is odd. (A circulant matrix is a matrix in which each row is rotated one element to the right relative to the preceding row.) In general, a circulant matrix is completely specified by its first row. However, since H_s is a symmetric circulant matrix, it is completely specified by the first $M + 1$ entries of the first row. Let the first entry of the first row be 0, and the next M entries of the first row be $s_1, s_2, \ldots, s_M$. In other words, the first $M + 1$ entries of the first row of H_s are 0 followed by the string s. Then the remaining entries of the first row are $s_M, s_{M-1}, \ldots, s_1$.

Given a black box for the entries of s, we can easily construct a black box for the entries of H_s. Indeed, one query to H can be simulated with at most one query to the string s. Sometimes no query to s is needed, since the diagonal entries of H_s are always 0.

Since H_s is a circulant matrix, it is diagonalized by the discrete Fourier transform. Its eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{N-1}$ are

$$\lambda_k = 2 \sum_{j=1}^{M} s_j \cos\left(\frac{2\pi jk}{N}\right), \quad \text{and in particular,} \quad \lambda_0 = 2 \sum_{j=1}^{M} s_j. \tag{18}$$

Thus the time evolution of H_s can be used to learn whether $\sum_j s_j$ is $-\sqrt{M \log M}$ or $+\sqrt{M \log M}$. Since $\lambda_0 = 2 \sum_j s_j$, the two cases can be distinguished by determining the sign of λ_0. Note that we

know the eigenvector corresponding to λ_0: it is the first column of the discrete Fourier transform matrix, i.e., the uniform superposition over all computational basis states.

Consider the eigenvalues and eigenvectors of the unitary matrix $e^{-iH\tau}$ that corresponds to evolving H for time $\tau = \pi/4\sqrt{M\log M}$. The eigenvectors of this matrix are the same as those of H, and each eigenvalue λ_k of H corresponds to the eigenvalue $\exp(-i\lambda_k\tau)$ of $e^{-iH\tau}$. Thus the uniform superposition is an eigenvector of $e^{-iH\tau}$ with eigenvalue $\exp(-i\lambda_0\tau) = \exp\left(-i\pi\sum_i s_i/2\sqrt{M\log M}\right)$. Since $\sum_i s_i/\sqrt{M\log M}$ is either ± 1, the two possible eigenvalues are $\pm i$. Because the eigenvector is known, the two possibilities can be easily distinguished by phase estimation on the unitary $e^{-iH\tau}$. Since the problem of distinguishing these two cases has an $\Omega(\sqrt{M/\log M})$ average-case lower bound by Lemma 2, we get an $\Omega(\sqrt{M/\log M})$ average-case lower bound for simulating such Hamiltonians for time $\tau = \pi/4\sqrt{M\log M}$.

Now we want to show that a $\mathrm{poly}(\|Ht\|, \log N)$ Hamiltonian simulation algorithm violates this lower bound. To do this, we need to know the typical behavior of $\|H_s\|$ when s satisfies the promise. Let $\mathcal{S} := \{-1, +1\}^M$ and let $\mathcal{P}$ be the subset of strings in $\mathcal{S}$ that satisfy the promise. As a first step, let us see the behavior of $\|H_s\|$ for all strings $s \in \mathcal{S}$, not just those that satisfy the promise.

Lemma 3. *Let $H_s \in \mathbb{R}^{N \times N}$ be a symmetric circulant matrix of size $N = 2M + 1$ with the first $M + 1$ entries of the first row given by 0 followed by a string $s \in \mathcal{S}$. If s is chosen uniformly at random from $\mathcal{S}$, denoted $s \in_R \mathcal{S}$, then for any $d > 0$,*

$$\Pr_{s \in_R \mathcal{S}}\left(\|H_s\| \geq 4d\sqrt{M\log M}\right) \leq \frac{4 + o(1)}{M^{2d^2 - 1}}. \tag{19}$$

In fact, even stronger results of this kind are known [15, 20], but the above bound is easy to prove and sufficient for our purposes. Using Lemma 3, we wish to bound the spectral norm of H_s when $s \in_R \mathcal{P}$. We can do so by first calculating the probability that a randomly selected string satisfies the promise.

Lemma 4. *If $s \in_R \mathcal{S}$, then the probability that s satisfies the promise is*

$$\Pr_{s \in_R \mathcal{S}}(s \in \mathcal{P}) = \Theta(1/M). \tag{20}$$

Using Lemmas 3 and 4, we can upper bound the probability that $\|H_s\|$ is large when s is chosen uniformly at random from $\mathcal{P}$. If X is the event that $\|H_s\| \geq 4d\sqrt{M\log M}$ and Y is the event that $s \in \mathcal{P}$, then $\Pr(X)$ is given by Lemma 3 and $\Pr(Y)$ is given by Lemma 4. In these terms, we can compute an upper bound for $\Pr(X|Y)$ as follows:

$$\Pr_{s \in_R \mathcal{P}}\left(\|H_s\| \geq 4d\sqrt{M\log M}\right) = \Pr(X|Y) = \frac{\Pr(X \cap Y)}{\Pr(Y)} \leq \frac{\Pr(X)}{\Pr(Y)} = O(M^{2 - 2d^2}). \tag{21}$$

To achieve a contradiction, assume that for some constant $c > 0$, there exists a Hamiltonian simulation algorithm using $O\left((\|Ht\|\log M)^c\right)$ queries to simulate H for time t. By (21), we know that H_s almost always has spectral norm smaller than $4d\sqrt{M\log M}$ when $s \in_R \mathcal{P}$. Since $\|H_s\| \leq \|H_s\|_1 = 2M$, we can compute the average-case query complexity of the claimed algorithm as follows:

$$\mathbb{E}_{s \in_R \mathcal{P}}(\|Ht\|^c) \leq \Pr_{s \in_R \mathcal{P}}\left(\|H_s\| < 4d\sqrt{M\log M}\right) O\left(\left(4d\sqrt{M\log M}\frac{\pi}{4\sqrt{M\log M}}\log M\right)^c\right)$$

$$+ \Pr_{s \in_R \mathcal{P}}\left(\|H_s\| \geq 4d\sqrt{M\log M}\right) O\left(\left((2M)\frac{\pi}{4\sqrt{M\log M}}\log M\right)^c\right) \tag{22}$$

$$\leq O((d\log M)^c) + O\left(M^{c/2 - 2d^2 + 2}(\log M)^{c/2}\right), \tag{23}$$

and by choosing $2d^2 > c/2 + 2$, we have

$$\mathop{\mathbb{E}}_{s \in_{\mathrm{R}} \mathcal{P}} (\|Ht\|^c) = O\left((\log M)^c\right). \tag{24}$$

Thus the average-case query complexity of the claimed algorithm is $O\left((\log M)^c\right)$, which violates the lower bound of $\Omega(\sqrt{M/\log M})$. $\qquad\square$

The proof technique above can be extended to rule out algorithms with query complexity sub-exponential in $(\|Ht\|, \log N)$ as well, by changing the promised set (i.e., the value of B used in Lemma 2) and choosing a larger value of d in Lemma 3. Exponential functions of $(\|Ht\|, \log N)$ cannot be ruled out, of course, since any Hamiltonian can be simulated by making $O(N^2)$ queries, which is exponential in $\log N$. On the other hand, if we insist that the query complexity of an algorithm depends only on $\|Ht\|$ (and not $\log N$), then the proof above can be modified to rule out algorithms whose time complexity is an arbitrary function of $\|Ht\|$. For example, there exists no Hamiltonian simulation algorithm that makes $\exp(\exp(\|Ht\|))$ queries.

Finally, we emphasize that even though the above proof involves average-case complexity and distributions over inputs, Theorem 4 is a statement about the worst-case complexity of simulating Hamiltonians.

5 Simulation complexity for structured Hamiltonians

As the previous section shows, we cannot hope for general Hamiltonian simulation algorithms that scale polynomially in the spectral norm of the Hamiltonian. Although we do know algorithms that scale like $O(\|\mathrm{abs}(H)t\|)$, Lemma 1 tells us that $\|\mathrm{abs}(H)\|$ could be exponentially larger than $\|H\|$. However, we can achieve better scaling for special classes of Hamiltonians. For example, we saw in Section 2 that much stronger bounds hold for sparse Hamiltonians.

We can also improve the inequalities of Lemma 1 for certain classes of non-sparse Hamiltonians. For example, consider the class of Hamiltonians whose graphs are trees (where the graph of a matrix refers to the graph of its nonzero entries). Such Hamiltonians can be efficiently simulated even when they are not sparse: in this case, Theorem 2 gives a simulation using $O(\|Ht\|)$ steps of a discrete-time quantum walk, because when the graph of H is a tree, $\|\mathrm{abs}(H)\| = \|H\|$.

Proposition 1. *If the graph of a Hermitian matrix H is a tree, then there exists a unitary matrix U such that $UHU^\dagger = \mathrm{abs}(H)$. In particular, $\|\mathrm{abs}(H)\| = \|H\|$.*

Proof. The matrix U is diagonal. To define U_{ii}, we arbitrarily fix some vertex as the root and consider the unique path from the root to vertex i. Let the path contain the vertices $i_0, i_1, \ldots, i_{p-1}, i_p, i$, where i_0 is the root and i_p is the parent of i. For each nonzero entry of H, define $\alpha_{ij} := H_{ij}/|H_{ij}|$. Then let $U_{ii} := 1$ if i is the root and $U_{ii} := \alpha_{i_0 i_1} \alpha_{i_1 i_2} \cdots \alpha_{i_{p-1} i_p} \alpha_{i_p i}$ otherwise.

Since U is diagonal, $(UHU^\dagger)_{ij} = U_{ii} H_{ij} U_{jj}^*$. If i and j are not adjacent in the tree, then $(UHU^\dagger)_{ij} = H_{ij} = 0$ as required. Otherwise, suppose without loss of generality that j is the parent of i. Then $U_{ii} U_{jj}^* = |\alpha_{i_0 i_1}|^2 |\alpha_{i_1 i_2}|^2 \cdots |\alpha_{i_{p-1} i_p}|^2 \alpha_{ji} = H_{ji}/|H_{ij}|$, so $(UHU^\dagger)_{ij} = |H_{ij}|$ as claimed. $\qquad\square$

Thus Theorem 2 gives a simulation using $O(\|Ht\|)$ steps of a discrete-time quantum walk. However, there is another simulation method for such Hamiltonians that uses only $\mathrm{mcn}(Ht)^{1+o(1)}$ steps [7, Theorem 4]. It seems from Lemma 1 that the former method might be inferior, but due to Proposition 2 below it is, in fact, superior to the $\mathrm{mcn}(Ht)^{1+o(1)}$ simulation (except with respect to error scaling), since $\|Ht\| \leq 2\,\mathrm{mcn}(Ht)$ when the graph of the Hamiltonian H is a tree.

If H can be expressed as the sum of a small number of Hamiltonians, each of whose graph is a forest, then H can be efficiently simulated when $\|H\|$ is small. Recall that a graph is said to have arboricity k if its adjacency matrix can be written as the sum of the adjacency matrices of k forests, but not $k-1$ forests.

Proposition 2. *If the graph of a Hamiltonian H has arboricity k, then $\|\text{abs}(H)\| \leq 2k\,\text{mcn}(H)$. Moreover, $\|\text{abs}(H)\| \leq 2k\|H\|$ and $\|H\| \leq 2k\,\text{mcn}(H)$.*

Proof. We begin by considering the case of a star graph. A star graph is a tree on n vertices with one vertex having degree $n-1$ and the others having degree 1 (i.e., the complete bipartite graph $K_{1,n-1}$). We show that if S is a Hamiltonian whose graph is a star,

$$\text{mcn}(S) = \|S\| = \|\text{abs}(S)\|. \tag{25}$$

By permuting the vertices, the first vertex can be chosen to be the one with maximum degree. Now the first column of the matrix S completely determines the Hamiltonian. Let the first column be w. The matrix S has first column w and first row $w^\dagger$. It is easy to see that $\text{mcn}(S) = \|w\|$. S has exactly two nonzero eigenvalues, $\pm\|w\|$, corresponding to the eigenvectors $\|w\|e_1 \pm w$, where e_1 is the first column of the identity matrix. Since $\|S\|$ is the maximum eigenvalue, $\|S\| = \text{mcn}(S)$.

Moreover, since $\text{abs}(S)$ is a Hamiltonian whose graph is a star, we have $\|\text{abs}(S)\| = \text{mcn}(\text{abs}(S))$. For any matrix H, $\text{mcn}(\text{abs}(H)) = \text{mcn}(H)$, since the norms of the columns depend only on the magnitude of each entry. This proves the desired result, $\text{mcn}(S) = \|S\| = \|\text{abs}(S)\|$. These results also hold for a disjoint union of star graphs (a forest of stars), since the above norms all have the property that $\nu(A_1 \oplus \ldots \oplus A_n) = \max(\nu(A_1), \ldots, \nu(A_n))$.

To show the result for graphs of arboricity k, we begin by showing how a rooted tree can be decomposed into the sum of two forests of stars. The first forest contains all the edges in which the parent vertex is at a even distance from the root. The second forest contains the rest of the edges. This decomposes a rooted tree into two forests of stars, and similarly decomposes a forest into two forests of stars. Since the Hamiltonian has arboricity k, it can be decomposed into k forests, which can be decomposed into $2k$ forests of stars.

Thus $H = \sum_{l=0}^{2k} S_l$, where each of the S_l is a Hermitian matrix whose graph is a forest of stars. Moreover, the matrices S_l have no overlapping edges, i.e., if $(S_l)_{ij} \neq 0$ for some l, then $(S_l)_{ij} = 0$ for all other l. Therefore, for all i, j, l, $H_{ij} \geq (S_l)_{ij}$, which implies $\text{mcn}(H) \geq \text{mcn}(S_l)$ for all l. This gives

$$\text{mcn}(H) \geq \frac{1}{2k}\sum_l \text{mcn}(S_l) = \frac{1}{2k}\sum_l \|S_l\|. \tag{26}$$

Using the triangle inequality, we find

$$\|\text{abs}(H)\| \leq \sum_l \|\text{abs}(S_l)\| = \sum_l \|S_l\|. \tag{27}$$

Combining (26) and (27) gives the main result. Using Lemma 1 and the main result gives $\|\text{abs}(H)\| \leq 2k\|H\|$ and $\|H\| \leq 2k\,\text{mcn}(H)$. $\qquad\square$

6   Open questions

Although we have ruled out the possibility of a generic Hamiltonian simulation algorithm using only $\text{poly}(\|Ht\|, \log N)$ operations, we can nevertheless hope that some nontrivial classes of Hamiltonians can be simulated in $\text{poly}(\|Ht\|, \log N)$ steps even though $\|H\| \ll \|\text{abs}(H)\|$.

One approach is to consider changing the basis in which the Hamiltonian is simulated. Clearly, if unitary transformations U and $U^\dagger$ can be performed efficiently, then H can be simulated efficiently if and only if $UHU^\dagger$ can. There must exist bases in which $UHU^\dagger$ is sparse (such as the basis in which it is diagonal), which may lead to efficient simulations of H. Some trivial classes of Hamiltonians can be simulated in this way, such as Hamiltonians that are tensor products of small factors. For example, the Hamiltonian $R^{\otimes n}$, where $R := \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix} \right)/\sqrt{2}$ is the Hadamard matrix, has $\|R^{\otimes n}\| = 1$ and $\|R^{\otimes n}\|_1 = \|\mathrm{abs}(R^{\otimes n})\| = 2^{n/2}$, yet the evolution according to $R^{\otimes n}$ is easy to simulate. A similar simulation for a case where the Hamiltonian is not a tensor product was used in Ref. [9].

An alternative method is to investigate ways of decomposing a Hamiltonian as a sum of Hamiltonians that can be efficiently simulated. For example, we can simulate Hamiltonians whose graphs have polynomial arboricity by decomposing them into stars (although we saw in Proposition 2 that such Hamiltonians can already by simulated efficiently by the method of Ref. [7] since they satisfy $\|\mathrm{abs}(H)\| = \mathrm{poly}(\|H\|)$). More generally, other graph decompositions could give rise to new efficient simulations.

Another interesting problem is to find classes of Hamiltonians that can be simulated in sublinear time. These correspond to quantum systems whose time evolution can be fast-forwarded. Some Hamiltonians may even be simulated in constant time, if $e^{-iH\tau} = I$ after some constant time τ (for example, the case $R^{\otimes n}$ mentioned above has $\tau = 2\pi$).

Acknowledgments

Appendix: Proofs of lemmas

In this appendix, we prove Lemmas 2, 3, and 4.

Lemma 2. *Suppose we are given black-box access to a string $s \in \{-1, +1\}^M$, where s is chosen uniformly at random from the set of strings with $\sum_i s_i \in \{-B, +B\}$. Then determining $\sum_i s_i$ has average-case quantum query complexity $\Theta(M/B)$.*

Proof. We show the lower bound by first showing the same lower bound for the worst-case problem using the quantum adversary method [2] and then reducing the worst-case problem to the average-case problem.

For the worst-case lower bound, we use the notation of Theorem 2 of Ref. [2]. We require two sets of inputs X and Y that have different outputs. Let X be the set of all strings for which $\sum_i s_i = -B$, and Y be the set for which $\sum_i s_i = +B$. We define a relation between the sets as follows. Let an element $x \in X$ be related to an element of $y \in Y$ if and only if y can be reached from x by changing exactly $B/2$ -1s to $+1$s in the string x. Note that a string in X has exactly $(M/2 + B/2)$ -1s and $(M/2 - B/2)$ $+1$s.

Using these sets X and Y, and the relation defined above, it is easy to see that

$$m = m' = \binom{M/2 + B/2}{B} \quad \text{and} \quad l = l' = \binom{M/2 + B/2 - 1}{B - 1}, \tag{28}$$

so

$$\frac{m}{l} = \frac{m'}{l'} = \frac{1}{2}\left(\frac{M}{B} + 1\right). \tag{29}$$

The adversary method now provides a lower bound of $\Omega\left(\sqrt{mm'/ll'}\right) = \Omega(M/B)$ for the worst-case query complexity of this problem.

The worst-case query complexity can now be reduced to the average-case query complexity under the uniform distribution over all input strings satisfying the promise. To do this, we first apply a uniformly random permutation to the input string, and then with probability $\frac{1}{2}$ multiply all the entries by -1 (and leave them unchanged with probability $\frac{1}{2}$). The resulting distribution is now uniform over all input strings satisfying the promise. If the string is not multiplied by -1, then the output of the permuted string is the same as the input string. If all the entries are multiplied by -1, then the output of the modified string is the opposite of that of the original input.

The lower bound is tight due to a matching upper bound provided by the algorithm for approximate quantum counting [5]. To distinguish the two types of inputs, we can approximately count the number of $+1$s to accuracy $\epsilon = B/2M$, which requires $O(M/B)$ queries. $\qquad\square$

Lemma 3. *Let $H_s \in \mathbb{R}^{N \times N}$ be a symmetric circulant matrix of size $N = 2M + 1$ with the first $M + 1$ entries of the first row given by 0 followed by a string $s \in \mathcal{S}$. If s is chosen uniformly at random from $\mathcal{S}$, denoted $s \in_{\mathrm{R}} \mathcal{S}$, then for any $d > 0$,*

$$\Pr_{s \in_{\mathrm{R}} \mathcal{S}} \left(\|H_s\| \geq 4d\sqrt{M \log M} \right) \leq \frac{4 + o(1)}{M^{2d^2 - 1}}. \tag{30}$$

Proof. The eigenvalues of H_s are $\lambda_r = 2\sum_{j=1}^{M} s_j \cos\frac{2\pi jr}{N}$, where $r \in \{0, 1, \ldots, N - 1\}$. We wish to bound the probability that λ_r is large, so as to bound the probability of $\|H_s\| = \max_r |\lambda_r|$ being large. This is achieved by applying Hoeffding's inequality [16, Theorem 2].

Theorem 5 (Hoeffding's inequality). *If $X_1, X_2, \ldots, X_M$ are independent and $a_j \leq X_j \leq b_j$ for all $1 \leq j \leq M$, then for any $t > 0$, we have*

$$\Pr\left(X - \mathbb{E}\left(X\right) \geq Mt\right) \leq \exp\left(\frac{-2M^2 t^2}{\sum_{j=1}^{M}(b_j - a_j)^2}\right) \tag{31}$$

where $X = \sum_{j=1}^{M} X_j$.

If we take $X_j = 2s_j \cos\frac{2\pi jr}{N}$, then $X = \lambda_r = 2\sum_{j=1}^{M} s_j \cos\frac{2\pi jr}{N}$, each X_j is between -2 and $+2$, and $\mathbb{E}\left(X\right) = \sum_{j=1}^{M} \mathbb{E}\left(X_j\right) = 0$. By choosing $t = 4d\sqrt{\log M/M}$, we get

$$\Pr\left(\lambda_r \geq 4d\sqrt{M \log M}\right) \leq \exp\left(\frac{-2M^2(16d^2 \log M/M)}{\sum_{j=1}^{M} 4^2}\right) = \frac{1}{M^{2d^2}}. \tag{32}$$

Since a similar inequality holds when X_j is replaced by $-X_j$, we get

$$\Pr\left(|\lambda_r| \geq 4d\sqrt{M \log M}\right) \leq \frac{2}{M^{2d^2}}. \tag{33}$$

Finally, since $\|H_s\| = \max_r |\lambda_r|$, a union bound gives

$$\Pr\left(\|H_s\| \geq 4d\sqrt{M \log M}\right) \leq \frac{2N}{M^{2d^2}}, \tag{34}$$

which implies the desired result. $\qquad\square$

Lemma 4. *If $s \in_R \mathcal{S}$, then the probability that s satisfies the promise is*

$$\Pr_{s \in_R \mathcal{S}}(s \in \mathcal{P}) = \Theta(1/M). \tag{35}$$

Proof. Of the 2^M strings of length M, those with sum $-\sqrt{M \log M}$ or $+\sqrt{M \log M}$ have either $\frac{1}{2}(M + \sqrt{M \log M})$ +1s or $\frac{1}{2}(M + \sqrt{M \log M})$ −1s. Thus the total number of such strings is

$$2\binom{M}{\frac{M + \sqrt{M \log M}}{2}}. \tag{36}$$

We can asymptotically approximate this expression using a well-known approximation for the binomial coefficients (see for example equations 4.5 and 4.10 of Ref. [19]), which states that

$$\binom{n}{k} \sim \frac{2^n \exp\left(-2(k - n/2)^2/n\right)}{\sqrt{\pi n/2}} \tag{37}$$

provided $|k - n/2| = o(n^{2/3})$. Applying this to (36), we get

$$2\binom{M}{\frac{M + \sqrt{M \log M}}{2}} = \Theta\left(2^M \exp(-\log M/2)/\sqrt{M}\right) = \Theta(2^M/M), \tag{38}$$

which proves the claim. $\qquad\square$

References

[1] D. Aharonov and A. Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, Proc. 35th ACM Symposium on Theory of Computing, 2003, pp. 20–29, available at quant-ph/0301023.

[2] A. Ambainis, *Quantum lower bounds by quantum arguments*, J. Comput. Syst. Sci. **64** (2002), no. 4, 750–767, available at quant-ph/0002066. Preliminary version in STOC 2000.

[3] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, *Quantum lower bounds by polynomials*, J. ACM **48** (2001), no. 4, 778–797, available at quant-ph/9802049. Preliminary version in FOCS 1998.

[4] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, *Efficient quantum algorithms for simulating sparse Hamiltonians*, Commun. Math. Phys. **270** (2007), no. 2, 359–371, available at quant-ph/0508139.

[5] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, *Quantum amplitude amplification and estimation*, Quantum Computation and Information, Contemp. Math., vol. 305, AMS, 2002, pp. 53–74, available at quant-ph/0005055.

[6] A. M. Childs, *Quantum information processing in continuous time*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[7] _____, *On the relationship between continuous- and discrete-time quantum walk*, to appear in Commun. Math. Phys., available at arXiv:0810.0312.

[8] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Exponential algorithmic speedup by quantum walk*, Proc. 35th ACM Symposium on Theory of Computing, 2003, pp. 59–68, available at quant-ph/0209131.

[9] A. M. Childs, L. J. Schulman, and U. V. Vazirani, *Quantum algorithms for hidden nonlinear structures*, Proc. 48th IEEE Symposium on Foundations of Computer Science, 2007, pp. 395–404, available at arXiv:0705.2784.

[10] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum algorithm for the Hamiltonian NAND tree*, Theory of Computing **4** (2008), no. 1, 169–190, available at quant-ph/0702144.

[11] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *Limit on the speed of quantum computation in determining parity*, Phys. Rev. Lett. **81** (1998), no. 24, 5442–5444, available at quant-ph/9802045.

[12] _____, *Quantum computation by adiabatic evolution*, quant-ph/0001106.

[13] E. Farhi and S. Gutmann, *Analog analogue of a digital quantum computation*, Phys. Rev. A **57** (1998), 2403–2406, available at quant-ph/9612026.

[14] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21** (1982), 467–488.

[15] G. Halasz, *On the result of Salem and Zygmund concerning random polynomials*, Studia Sci. Math. Hung. **8** (1973), 369–377.

[16] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc. **58** (1963), 13–30.

[17] E. Knill, *Approximation by quantum circuits*, Technical Report LAUR-95-2225, Los Alamos National Laboratory, quant-ph/9508006.

[18] S. Lloyd, *Universal quantum simulators*, Science **273** (1996), 1073–1078.

[19] A. M. Odlyzko, *Asymptotic enumeration methods*, Handbook of Combinatorics (R. L. Graham, M. Groetschel, and L. Lovasz, eds.), Vol. 2, MIT Press, 1995, pp. 1063–1229.

[20] R. Salem and A. Zygmund, *Some properties of trigonometric series whose terms have random signs*, J. Acta. Math. **91** (1954), 245–301.