Quantum Information and Computation, Vol. 16, No. 15&16 (2016) 1295–1317

CORRECTED QUANTUM WALK FOR OPTIMAL HAMILTONIAN SIMULATION

DOMINIC W. BERRY
Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia

LEONARDO NOVO
Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia
Physics of Information and Quantum Technologies Group, Instituto de Telecomunicações, Lisbon, Portugal
Instituto Superior Técnico, Universidade de Lisboa, Portugal

Received June 18, 2016
Revised September 1, 2016

We describe a method to simulate Hamiltonian evolution on a quantum computer by repeatedly using a superposition of steps of a quantum walk, then applying a correction to the weightings for the numbers of steps of the quantum walk. This correction enables us to obtain complexity which is the same as the lower bound up to double-logarithmic factors for all parameter regimes. The scaling of the query complexity is $O\left(\tau \frac{\log\log\tau}{\log\log\log\tau} + \log(1/\epsilon)\right)$ where $\tau := t\|H\|_{\max} d$, for ϵ the allowable error, t the time, $\|H\|_{\max}$ the max-norm of the Hamiltonian, and d the sparseness. This technique should also be useful for improving the scaling of the Taylor series approach to simulation, which is relevant to applications such as quantum chemistry.

Keywords: Quantum algorithms, quantum query complexity, Hamiltonian simulation, quantum walk
Communicated by: R Cleve & G Milburn

1 Introduction

The simulation of physical quantum systems is a natural application where quantum computers can achieve an exponential speedup (in the dimension of the system), and was Feynman's original motivation for proposing quantum computers [1]. An algorithm for the case of a physical system composed of low-dimensional subsystems and a Hamiltonian that is a sum of interaction Hamiltonians was proposed by Lloyd [2]. An alternative scenario is that where the matrix representing the Hamiltonian is sparse, and there is a procedure for calculating the positions and values of nonzero entries, which can be regarded as an oracle [3]. The advantage of this approach is that it can be used to not just simulate Hamiltonians corresponding to physical systems, but to design other algorithms [4, 5, 6, 7]. There have been many papers providing improved algorithms for simulating sparse Hamiltonians [8, 9, 10, 11, 12, 13, 14, 15, 16].

One particular approach is that of a quantum walk [10, 12], where a step of the quantum walk can be implemented using an oracle for the Hamiltonian, and has eigenvalues related to that of the Hamiltonian. In Refs [10, 12], the technique was to use phase estimation to estimate the eigenvalue of the quantum walk step, and use that to apply the appropriate

eigenvalue for Hamiltonian evolution. Another common approach is that based on a Lie-Trotter product [2, 3, 8, 9, 11]. A more recent improvement is to use a control qubit for each step in the product, and compress the control qubits [14]. Surprisingly, this turns out to be equivalent to an approach based on a Taylor expansion of the exponential for the Hamiltonian evolution [15]. These approaches provide a scaling that is logarithmic in $1/\epsilon$, where ϵ is the allowable error for the simulation, in contrast to previous approaches which were polynomial in $1/\epsilon$.

These results motivated an improved approach to quantum walks, where a superposition of different numbers of steps of the quantum walk is used [16]. This approach enables simulations with query complexity

$$O\left(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right), \tag{1}$$

where $\tau := t\|H\|_{\max} d$. Here t is the time that the Hamiltonian evolution is to be simulated over, $\|H\|_{\max}$ is the max-norm of the Hamiltonian, and d is the sparseness of the Hamiltonian, which is the maximum number of nonzero elements in any row or column. The query complexity corresponds to the number of calls to the oracle for elements of the Hamiltonian. In contrast, the lower bound to the complexity is [16]

$$\Omega\left(\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right). \tag{2}$$

Hence, although there is optimal scaling (up to a logarithmic factor) in any of the parameters individually, the complexity of the algorithm has a product, whereas the lower bound has a sum, so there is room to improve the complexity in the parameter regime where τ is close to $\log(1/\epsilon)$. Note that there is also a complexity in terms of the number of additional gates. We do not consider that complexity here.

Here we show a method to reduce the complexity to close to the lower bound. The idea is to use the superposition of different numbers of steps of the quantum walk as in [16], but correct the weightings. The scaling of the query complexity for our approach is

$$O\left(\tau \frac{\log\log\tau}{\log\log\log\tau} + \log(1/\epsilon)\right). \tag{3}$$

Our technique is flexible enough that it can be applied to the Taylor series approach to quantum walks as well (though we will not provide the proof here). The Taylor series approach is important because it can be used for situations like quantum chemistry [17, 18], where the Hamiltonian naturally decomposes into a sum of terms, but it is difficult to construct an oracle to give elements of the Hamiltonian. Another advantage of the Taylor series approach is that it gives better scaling for the number of additional gates (the above scalings are the query complexity). After completion of this work, we were made aware of another approach that yields complexity slightly closer to the lower bound [19], but is only applicable to the quantum walk approach, not the Taylor series approach.

In the next Section we provide the theoretical background of the techniques from Ref. [16]. In Section 3 we provide a description of our algorithm, and introduce definitions required in later sections. Then in Section 4 we define algorithms with a single round of correction and with two rounds of correction. We provide the proofs for a single round of correction and two rounds of correction in Sections 5 and 6, respectively. We conclude in Section 7.

2 Background

In the Hamiltonian simulation problem, a Hamiltonian H acting on n qubits is given, as well as a time t and maximum allowable error $\epsilon > 0$, and the task is to implement the unitary operation e^{-iHt} within error ϵ. The error may be quantified by the diamond norm distance, although as shown in [16] it is equivalent to consider the spectral norm of the difference of the operators. The question of appropriate measurements to perform on the resulting quantum state is separate from the Hamiltonian simulation problem. For sparse Hamiltonian simulation, the Hamiltonian has a matrix representation in the computational basis with no more than d nonzero entries in any row or column. The Hamiltonian is specified by two oracles. One takes as input a row number j and integer ℓ, and outputs the position of the ℓth nonzero element in row j. This oracle computes the value in place, so it acts as

$$O_F|j,\ell\rangle = |j, f(j,\ell)\rangle, \tag{4}$$

where $f(j,\ell)$ is the function giving the column index. The other oracle takes as input the row and column number for the Hamiltonian, and outputs an encoding of the value of corresponding matrix element of H. This oracle acts as

$$O_H|j,k,z\rangle = |j,k,z\oplus H_{jk}\rangle, \tag{5}$$

where z is a bit string used to represent entries of H, and $\oplus$ indicates bitwise addition. These oracles represent some procedure for calculating the positions and values of nonzero entries in the Hamiltonian.

Next we summarise the key results needed from [10, 12, 16]. Those works consider a quantum walk which is based on controlled state preparation. The Hilbert space is expanded from $\mathbb{C}^{\mathcal{N}}$, where $\mathcal{N} = 2^n$, to $\mathbb{C}^{2\mathcal{N}} \otimes \mathbb{C}^{2\mathcal{N}}$. First an ancilla qubit is appended, which expands the space to $\mathbb{C}^{2\mathcal{N}}$, then the original Hilbert space and ancilla qubit are duplicated. The duplication is performed using a controlled state preparation operator [16]

$$T := \sum_{j=1}^{\mathcal{N}} \sum_{b\in\{0,1\}} (|j\rangle\langle j| \otimes |b\rangle\langle b|) \otimes |\varphi_{jb}\rangle, \tag{6}$$

where $|\varphi_{j1}\rangle = |0\rangle|1\rangle$ and

$$|\varphi_{j0}\rangle = \frac{1}{\sqrt{d}} \sum_{\ell\in F_j} |\ell\rangle \left[\sqrt{\frac{H_{j\ell}^*}{X}}|0\rangle + \sqrt{1-\frac{|H_{j\ell}|}{X}}|1\rangle\right], \tag{7}$$

where F_j is the set of indices given by O_F on input j, and $X \geq \|H\|_{\max}$. For the algorithms described here, we will just take $X = \|H\|_{\max}$. The method to perform the controlled state preparation is described in Lemma 4 of Ref. [12]. The key feature of it that we need here is that it uses $O(1)$ calls to the oracles O_F and O_H.

We use the convention that the basis states are given in the order $|j_1\rangle|b_1\rangle|j_2\rangle|b_2\rangle$, where $|j_1\rangle$ and $|j_2\rangle$ are the original Hilbert space and duplicated space, and $|b_1\rangle$ and $|b_2\rangle$ are the orignal ancilla qubit and duplicated ancilla qubit. The quantum walk step is constructed from this controlled state preparation via

$$U := iS(2TT^\dagger - \mathbb{1}), \tag{8}$$

where S swaps the registers, as $S|j_1\rangle|b_1\rangle|j_2\rangle|b_2\rangle = |j_2\rangle|b_2\rangle|j_1\rangle|b_1\rangle$. All algorithms of this type work by starting in the initial Hilbert space, performing T to map the state into two copies of the Hilbert space, performing steps of the quantum walk U on these two subsystems, then using $T^\dagger$ to map the system back to the original Hilbert space.

The crucial feature of the step U is that its eigenvalues and eigenstates are related to those of H. In particular, an initial eigenstate $|\lambda\rangle$ of H will be mapped under T to a superposition of two eigenstates of U, $|\mu_+\rangle$ and $|\mu_-\rangle$. If the eigenvalue of the Hamiltonian is λ, the corresponding eigenvalues of U are

$$\mu_\pm = \pm e^{\pm i \arcsin(\lambda/Xd)}. \tag{9}$$

Using this expression and the generating function for Bessel functions, it can be shown that [16]

$$\sum_{m=-\infty}^{\infty} J_m(-tXd)\mu_\pm^m = e^{-i\lambda t}. \tag{10}$$

This expression effectively means that there is the equivalence

$$\sum_{m=-\infty}^{\infty} J_m(-tXd)U^m \equiv e^{-iHt}, \tag{11}$$

so if it is possible to apply a sum of powers of the quantum walk step U, it is possible to effectively apply the evolution under the Hamiltonian. Note that the Hamiltonian acts on a different space than U, so these operators are not equal. One needs to map from the original Hilbert space to the two copies of the space, apply the superposition of powers of U, then map back to the original Hilbert space to obtain e^{-iHt}.

The method to apply a linear combination of unitaries of this form is described in detail in Section IIIB of Ref. [16]. We summarise it for the case of powers of U here. First, rather than attempting to obtain the evolution for the complete time t, we divide the time up into r segments, and apply the linear combination of unitaries for each time t/r. Second, the infinite sum in Eq. (10) needs to be truncated to finite values. We denote the cutoff by M. The cutoff needs to be chosen sufficiently large that the error for the complete simulation due to the discarded terms is no greater than ϵ. If we were to attempt to apply the sum of unitaries given in Eq. (10), but for time t/r, the weightings would be $v_m = J_m(-tXd/r)$. The operation that we wish to perform can then be written as

$$\tilde{V} = \sum_{m=-M}^{M} v_m U^m. \tag{12}$$

To perform the linear combination of unitaries, first an ancilla is prepared in the state

$$|\chi\rangle = \frac{1}{\sqrt{s}} \sum_{m=-M}^{M} \sqrt{v_m}|m\rangle, \tag{13}$$

where $s = \sum_{m=-M}^{M} |v_m|$. Preparation of this ancilla requires no queries, and only $O(M)$ gates. Then one performs a controlled operation

$$\text{select}(U) := \sum_{m=-M}^{M} |m\rangle\langle m| \otimes U^m. \tag{14}$$

Then, given an initial state $|\psi\rangle$, the state after appending the ancilla in the state $|\chi\rangle$ and applying select(U) is

$$\text{select}(U)|\chi\rangle|\psi\rangle = \frac{1}{\sqrt{s}} \sum_{m=-M}^{M} \sqrt{v_m}|m\rangle U^m|\psi\rangle. \tag{15}$$

Next, if one were to apply the projector $|\chi\rangle\langle\chi|$, corresponding to measuring the ancilla as being in the state $|\chi\rangle$, the resulting state would be

$$|\chi\rangle \frac{1}{s}\tilde{V}|\psi\rangle, \tag{16}$$

where the normalization indicates the probability of success. That is, the desired sum of unitaries has been performed, but the probability of success is only about $1/s^2$.

An alternative way of viewing the measurement is that one first inverts the state preparation on the ancilla. That will then yield the state

$$|0\rangle \otimes \frac{1}{s}\tilde{V}|\psi\rangle + |\perp\rangle, \tag{17}$$

where $|0\rangle$ is the zero initial state for the ancilla, and $|\perp\rangle$ is a possibly entangled state between the ancilla and target system that is perpendicular to $|0\rangle$ on the ancilla. If one were to perform a measurement on the ancilla and obtain the result 0, that would correspond to applying the projector $|\chi\rangle\langle\chi|$ to the state in Eq. (15).

Instead of just measuring, one can perform a procedure called *oblivious amplitude amplification* (OAA) to boost the amplitude of the first term in Eq. (17) to close to 1. Define W_χ to be the operation corresponding to preparing the state $|\chi\rangle$ from $|0\rangle$, then applying select(U), then inverting the state preparation operation. See Lemma 5 of Ref. [16] for an explanation of this operation (the notation used in [16] is W instead of W_χ). Also define $P := |0\rangle\langle 0| \otimes \mathbb{1}$ to be the projection onto $|0\rangle$ on the ancilla. Then in a single step of OAA, one reflects about the zero state on the ancilla via $\mathbb{1} - 2P$, performs $W_\chi^\dagger$, reflects about the zero state on the ancilla again, and performs W_χ. That is, the total operation performed for the segment is

$$-W_\chi(\mathbb{1} - 2P)W_\chi^\dagger(\mathbb{1} - 2P)W_\chi, \tag{18}$$

where the minus sign avoids a global phase factor. See Section 3.2 of Ref. [16] for a detailed explanation of OAA and the proof of accuracy given that the sum of unitaries is not exactly unitary. It is also possible to consider multiple steps of OAA, but here we only need a single step. The key feature of OAA that we need here is that after one step it gives the result

$$|0\rangle \otimes \left(\frac{3}{s}\tilde{V} - \frac{4}{s^3}\tilde{V}\tilde{V}^\dagger\tilde{V}\right)|\psi\rangle + |\perp'\rangle, \tag{19}$$

where $|\perp'\rangle$ is some new perpendicular state. If s is exactly 2, and $\tilde{V}$ is exactly unitary, then the state would be $|0\rangle\tilde{V}|\psi\rangle$, so the desired sum of unitaries has been performed. Even if $\tilde{V}$ is not exactly unitary, then the result will still be close to that desired. Also, if s is not exactly 2, but is instead less than 2, then it is trivial to include an extra ancilla that effectively increases it to 2, and the result is the same [16].

As the argument of the Bessel function is increased, it is necessary to increase M in order to limit the error. Then the value of s also increases (although not monotonically). This is why the evolution is divided up into r segments. By choosing r sufficiently large, $s \leq 2$, so only a single step of OAA is required for each segment. Then the overall Hamiltonian evolution is simulated by performing the simulation for each time t/r in succession. The overall complexity is then a product of the number of segments times the complexity for each segment. The value $s \leq 2$ can be obtained with the argument of the Bessel function $O(1)$. The argument of the Bessel function is $-t\,Xd/r$, so the total number of segments should be $r = O(tXd)$. Taking $X = \|H\|_{\max}$ and $\tau := t\|H\|_{\max}d$, the number of segments is $O(\tau)$.

In Ref. [16] the segments are just simulated in succession with no correction. Therefore, to bound the final error as no larger than ϵ, the error for each segment should be no larger than $O(\epsilon/\tau)$. The values of Bessel functions are bounded as [20, 10.14.4]

$$|J_m(z)| \leq \frac{1}{m!}\left|\frac{z}{2}\right|^{|m|}. \tag{20}$$

The error in the truncation will correspond to the absolute values of the Bessel functions which are omitted. If we truncate at $M > 0$, then the truncated terms will be bounded as [16]

$$2\sum_{m=M+1}^{\infty}|J_m(z)| \leq 4\frac{1}{(M+1)!}\left|\frac{z}{2}\right|^{M+1}. \tag{21}$$

Using this expression, the error for the segment can be limited to $O(\epsilon/\tau)$ if the scaling of the cutoff is

$$M = O\left(\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right). \tag{22}$$

Recall that each application of U uses $O(1)$ queries, and M is the cutoff to the maximum power of U, so the complexity for a single segment is $O(M)$. In Ref. [16] the overall complexity is then obtained by multiplying by the number of segments to give $O(\tau M)$, which gives the complexity scaling in Eq. (1).

3 New Algorithms with Corrections

The basis of our new technique is to allow a larger error for each segment, then reduce the error by performing a correction at the end. Because the simulation of the individual segments need not be as accurate, the cutoff on the Bessel functions can be smaller and need not depend on ϵ. That eliminates the multiplying factor depending on ϵ in the complexity. The correction at the end can use powers of U corresponding to the total complexity of the simulation thus far without changing the complexity except by a constant factor. Therefore the correction can greatly reduce the error.

The reason this approach works is that one step of OAA gives an effective operation which is (taking $s = 2$) [15]

$$V_{\text{oaa}} := \frac{3}{2}\tilde{V} - \frac{1}{2}\tilde{V}\tilde{V}^{\dagger}\tilde{V}, \tag{23}$$

where $\tilde{V}$ was defined in Eq. (12), and is the sum of powers of U used to approximate the Hamiltonian evolution for a single segment. What this means is that after performing the

OAA we have still performed an operation that is a sum of powers of U. To be more specific, the state has success flagged by zero in the ancilla, as well as a failure component, so is

$$|0\rangle \otimes V_{\text{oaa}}|\psi\rangle + |\perp'\rangle. \tag{24}$$

Also, the operation $\tilde{V}$ is

$$\tilde{V} := \sum_{m=-M}^{M} J_m(z) U^m, \tag{25}$$

where z satisfies

$$\sum_{m=-M}^{M} |J_m(z)| \leq 2. \tag{26}$$

For a single round of error correction, we wish to have $zr = -\tau$. If we select z to be the maximum value such that Eq. (26) is satisfied, then we would only be able to obtain a discrete set of values of τ (and therefore t). We instead take z to be the maximum value satisfying both (26) and $zr = -\tau$ for some integer r.

The effective operation after many segments is then a power of V_{oaa}, with success flagged by the $|0\rangle$ states in ancillas. That is, after r segments the state becomes

$$|0\rangle^{\otimes r} \otimes V_{\text{oaa}}^r|\psi\rangle + |\perp_2\rangle. \tag{27}$$

Hence, the effective operation is still a sum of powers of U. This sum of powers of U is something that we can easily calculate, and we can compare the weightings to those we would want in order to exactly simulate the Hamiltonian evolution. At the end we can apply another operation that is a sum of powers of U to correct the weightings, and reduce the error to order ϵ.

If the truncation of this sum of powers is of the same order as the maximum power of U for the operations that have been performed so far, then this correction will only give a constant multiplying factor to the complexity. What is more, this correction will have most of its weight on $m = 0$ (the identity), which means that the amplitude on success is high. More explicitly, define V_C to be the operation such that

$$V_C V_{\text{oaa}}^r = \sum_{m=-\infty}^{\infty} J_m(zr) U^m. \tag{28}$$

That is, it yields the desired weightings to give the evolution under the Hamiltonian. The correction has an expansion in terms of powers of U as

$$V_C = \sum_{m=-\infty}^{\infty} a_m U^m, \tag{29}$$

which we use to define the coefficients a_m. In practice we need to truncate the sum to limit the complexity of the correction, giving

$$\tilde{V}_C = \sum_{m=-N}^{N} a_m U^m. \tag{30}$$

Applying this approximate correction via an ancilla, in the same way as described above for $\tilde{V}$, gives the state

$$|0\rangle^{\otimes(r+1)} \otimes \frac{1}{s}\tilde{V}_C V_{\text{oaa}}^r|\psi\rangle + |\perp_3\rangle, \tag{31}$$

where $s = \sum_{m=-N}^{N} |a_m|$. Again, provided that $s \leq 2$, OAA can be used to obtain a result close to $\tilde{V}_C V_{\text{oaa}}^r|\psi\rangle$. The operation after OAA, denoted V'_{oaa}, is given by

$$V'_{\text{oaa}} = \frac{3}{2}\tilde{V}_C V_{\text{oaa}}^r - \frac{1}{2}\tilde{V}_C V_{\text{oaa}}^r (\tilde{V}_C V_{\text{oaa}}^r)^\dagger \tilde{V}_C V_{\text{oaa}}^r. \tag{32}$$

Again, the resulting state is flagged by zero states in the ancillas, and there is a contribution from an error state, in the same way as in Eqs. (17), (19), (24), (27), and (31). From now on we will not write these explicitly for brevity. Using Lemma 6 of [16], if $\tilde{V}_C V_{\text{oaa}}^r$ is within δ of a unitary matrix, then V'_{oaa} is within δ of $\tilde{V}_C V_{\text{oaa}}^r$.

In order to obtain $s \leq 2$ for the correction, we need the error in the segments to not build up above $O(1)$. This means that the error in the individual segments needs to be $O(1/\tau)$, so the cutoff is

$$M = O\left(\frac{\log \tau}{\log\log \tau}\right). \tag{33}$$

Multiplying by the number of segments, this gives a factor in the complexity of

$$O\left(\tau \frac{\log \tau}{\log\log \tau}\right). \tag{34}$$

Another limiting factor is that the final error needs to be $O(\epsilon)$. In the case where ϵ is small, it may be necessary to increase the cutoff N for the final correction. It is also convenient to increase M, though M does not need to scale with ϵ. It turns out that it is sufficient to choose $M\tau = O(\log(1/\epsilon))$, which gives the $O(\log(1/\epsilon))$ term in the final scaling.

The discussion so far is for a single round of error correction, which is addressed in detail in Section 5. It is also possible to repeatedly perform error correction, then perform a final round of error correction at the end. That is, we have the operation V'_{oaa} repeated r' times, with success again flagged by an ancilla state of $|0\rangle$. Then the new exact correction operation V'_C, is such that

$$V'_C (V'_{\text{oaa}})^{r'} = \sum_{m=-\infty}^{\infty} J_m(zrr')U^m. \tag{35}$$

Note that we now have r multiplied by r', so we should have $\tau = -zrr'$. For this case we need to select z such that Eq. (26) is satisfied and $\tau = -zrr'$, for appropriate choices of r and r'. The exact correction operation can again be expressed as a sum over powers of U as

$$V'_C = \sum_{m=-\infty}^{\infty} a'_m U^m, \tag{36}$$

which we use to define a'_m. The approximate correction operation is then the truncated form of this, with

$$\tilde{V}'_C = \sum_{m=-N'}^{N'} a'_m U^m. \tag{37}$$

This correction operation is applied using control registers and select(U) operations, then OAA is again applied.

Again we can estimate the complexity via a relatively simple approach. Take $r = \Theta(\log \tau)$, so the error after the first round of error correction (i.e., the error in $V'_{\rm oaa}$) is $O(1/\tau)$. Then we can take $r' = \Theta(\tau/\log \tau)$ to obtain $\tau = -zrr'$ as desired. With this value of r' the error will still be $O(1)$, so the final correction can be performed. Moreover, taking $r = \Theta(\log \tau)$, we can allow error for the individual segments (i.e., in $\tilde{V}$) to be $O(1/\log \tau)$. That means we can take the cutoff for $\tilde{V}$ to be

$$M = O\left(\frac{\log\log \tau}{\log\log\log \tau}\right). \tag{38}$$

That is what gives our double-log factor in the final result. The details of this double round of error correction are given in Section 6.

4 Summary of Algorithms

Algorithm 1: Hamiltonian simulation with a single round of correction

1. Apply the controlled state preparation operator T defined in Eq. (6) to map the state to two copies of the Hilbert space, as well as two ancilla qubits. The states $|\varphi_{jb}\rangle$ required for T are given via $|\varphi_{j1}\rangle = |0\rangle|1\rangle$ and Eq. (7).
2. For segments 1 to r with $r \in \Theta(\tau)$, perform the following steps.
   (a) Append an ancilla of dimension $2M + 1$ in state $|0\rangle$, and apply the operation to prepare the state $|\chi\rangle$ given in Eq. (13).
   (b) Perform the controlled unitary operation select(U) given in Eq. (14). The unitary operation U is defined in Eq. (8), and uses a swap operation S as well as the controlled preparation T.
   (c) Apply the inverse of the state preparation operation, giving the state in Eq. (17). That is, the operation $\tilde{V}$ has success flagged by a $|0\rangle$ state in the ancilla.
   (d) Apply a single step of OAA, as described in Lemma 5 of Ref. [16]. This yields the operation $V_{\rm oaa}$, as defined in Eq. (23), with success flagged by a zero in the ancilla.
3. Apply the correction $\tilde{V}_C$ defined in Eq. (30) via an ancilla and the unitary select(U), to give the state in Eq. (31).
4. Apply a single step of OAA on steps 2 and 3 above.
5. Invert the controlled state preparation operator T to map the state back into the original Hilbert space.

For the algorithm, we are provided an initial state $|\psi\rangle$ encoded on the qubits of the quantum computer. The complete algorithm with a single round of correction is given as Algorithm 1 above. At the end of this procedure, we have a state that is an approximation of $e^{-iHt}|\psi\rangle$. Provided M and N (the cutoffs for $\tilde{V}$ and $\tilde{V}_C$, respectively) are chosen appropriately, then the error will be within ϵ. For this algorithm we will always choose $M \geq 2$.

It is convenient for the discussion to refer to parts (a) to (d) of step 2 of Algorithm 1 as a *segment*. Then, we refer to the procedure in steps 2 to 4 of Algorithm 1 as a *compound segment*. The algorithm for two rounds of error correction repeats this compound segment a number of times denoted r'. The cutoff N for the correction $\tilde{V}_C$ is taken to be $3rM$, so it has the same complexity as performing V_{oaa}^r. Then a correction denoted V_C' is performed. The complete procedure is given as Algorithm 2 below.

Algorithm 2: Hamiltonian simulation with two rounds of correction

1. Apply the controlled state preparation operator T defined in Eq. (6) to map the state to two copies of the Hilbert space, as well as two ancilla qubits. The states $|\varphi_{jb}\rangle$ required for T are given via $|\varphi_{j1}\rangle = |0\rangle|1\rangle$ and Eq. (7).
2. For compound segments 1 to r' with $r' \in \Theta(\tau/\log\tau)$, perform the following steps.
   (a) Use step 2 of Algorithm 1 in order to apply V_{oaa}^r with $r \in \Theta(\log\tau)$.
   (b) Apply the correction $\tilde{V}_C$ via an ancilla and the controlled unitary select(U).
   (c) Apply a single step of OAA on steps 2 and 3 above.
3. Apply the correction $\tilde{V}_C'$ via an ancilla and the controlled unitary select(U).
4. Apply a single step of OAA on steps 2 and 3 above.
5. Invert the controlled state preparation operator T to map the state back into the original Hilbert space.

5 A Single Round of Error Correction

The specific results we derive for a single round of correction are as in the following Theorem.

Theorem 1 *A d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error $\epsilon > 0$ with*

$$O\left(\tau \frac{\log\tau}{\log\log\tau} + \log(1/\epsilon)\right) \tag{39}$$

queries, where $\tau := t\|H\|_{\max} d$.

To compare this result with the results of [16], the algorithm in that work had complexity

$$O\left(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right), \tag{40}$$

and a lower bound on the complexity

$$O\left(\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right). \tag{41}$$

These bounds differ most significantly in the regime where τ is comparable to $\log(1/\epsilon)$. In this regime the previous algorithm gives complexity approximately $O(\tau^2 \log\tau)$, whereas the complexity from Theorem 1 scales as $\tau\log\tau$, which is close to a square root improvement.

In order to prove this result, we first prove a number of intermediate lemmas. In these proofs we make a slight change from the notation used in the previous sections. We will now take the operations $\tilde{V}$, V_C, $\tilde{V}_C$, etc., explicitly as functions of U. This is because we wish to calculate s values for implementing a range of linear combinations of unitaries, and it is convenient to express s as a functional of the operation. In particular, for a function of the form $F(x) = \sum_n F_n x^n$, we define the functional s by

$$s(F) := \sum_n |F_n|. \tag{42}$$

First we provide a lemma on the elementary properties of the functional s.

Lemma 1 *The functional s satisfies the properties, for functions of U denoted F and G and α and β scalars,*

$$s(\alpha F + \beta G) \leq |\alpha| s(F) + |\beta| s(G) \tag{43}$$

$$s(FG) \leq s(F)s(G). \tag{44}$$

In addition, if F and G are sums over disjoint sets of powers, then

$$s(\alpha F + \beta G) = |\alpha| s(F) + |\beta| s(G). \tag{45}$$

Proof. To show the first result (43), for

$$F(x) = \sum_n F_n x^n, \qquad G(x) = \sum_n G_n x^n, \tag{46}$$

we have

$$\begin{aligned} s(\alpha F + \beta G) &= \sum_n |\alpha F_n + \beta G_n| \\ &\leq |\alpha| \sum_n |F_n| + \beta \sum_n |G_n| \\ &= |\alpha| s(F) + |\beta| s(G). \end{aligned} \tag{47}$$

To show Eq. (44),

$$(FG)(x) = \sum_n \sum_k F_{n-k} G_k x^n, \tag{48}$$

so

$$\begin{aligned} s(FG) &= \sum_n \left| \sum_k F_{n-k} G_k \right| \\ &\leq \sum_n \sum_k |F_{n-k}| \times |G_k| \\ &= \sum_n |F_n| \sum_m |G_m| \\ &= s(F)s(G). \end{aligned} \tag{49}$$

To show Eq. (45), denote the sets of powers for F and G by S_F and S_G, respectively, so that $S_F \cap S_G = \emptyset$. Then

$$\begin{aligned} s(\alpha F + \beta G) &= \sum_n |\alpha F_n + \beta G_n| \\ &= \sum_{n\in S_F} |\alpha F_n| + \sum_{n\in S_G} |\beta G_n| \\ &= |\alpha| \sum_n |F_n| + |\beta| \sum_n |G_n| \\ &= |\alpha| s(F) + |\beta| s(G) \end{aligned} \tag{50}$$

□.

The next lemma shows that the absolute values of the coefficients for the correction may be bounded. This result is needed in order to show that the correction only needs a single step of OAA.

Lemma 2 *In Algorithm 1, the coefficients a_m required for the correction operator $V_C(U)$ satisfy*

$$\sum_{m=-\infty}^{\infty} |a_m| \lesssim \left(1 - 2 \sum_{|m|>M} |J_m(z)|\right)^{-r}, \tag{51}$$

where $\lesssim$ indicates that higher-order terms in $\sum_{|m|>M} |J_m(z)|$ have been omitted.

Proof. For each segment we use a value z and a cutoff M, so the operation we attempt to perform is

$$\tilde{V}(U) := \sum_{m=-M}^{M} J_m(z) U^m. \tag{52}$$

This is equivalent to Eq. (25), except we have given $\tilde{V}$ as a function of U. The choice of z is restricted by Eq. (26), which ensures that $s(\tilde{V}) \le 2$. Similarly, we will write the operation that provides the exact evolution for that time interval as the function

$$V(U) = \sum_{m=-\infty}^{\infty} J_m(z) U^m. \tag{53}$$

Because $s(\tilde{V}) \le 2$, using a single step of OAA, the operation is $V_{\rm oaa}(U)$, with

$$V_{\rm oaa} := \frac{3}{2}\tilde{V} - \frac{1}{2}\tilde{V}\tilde{V}^\dagger \tilde{V}. \tag{54}$$

It is also convenient to define

$$\Delta := V - \tilde{V}. \tag{55}$$

After performing $V_{\rm oaa}(U)$ a number of times given by r, the actual operation is $[V_{\rm oaa}(U)]^r$, but the desired operation is $[V(U)]^r$. The correction V_C that will yield the exact operation is given implicitly by

$$V^r = V_C V_{\rm oaa}^r, \tag{56}$$

which is equivalent to Eq. (28). This means that we must have

$$\begin{aligned} V_C &= (V^\dagger V_{\mathrm{oaa}})^{-r} \\ &= \left(\frac{3}{2}V^\dagger\tilde{V} - \frac{1}{2}V^\dagger\tilde{V}(V^\dagger\tilde{V})^\dagger V^\dagger\tilde{V}\right)^{-r} \\ &= (\mathbb{1} - W)^{-r} \\ &= \sum_{k=0}^{\infty}\binom{r+k-1}{r-1}W^k, \end{aligned} \tag{57}$$

where

$$W = \frac{1}{2}\left(V^\dagger\Delta - \Delta^\dagger\tilde{V} + V^\dagger\Delta V^\dagger\Delta + V^\dagger\Delta\Delta^\dagger\tilde{V}\right) \tag{58}$$

$$= \frac{1}{2}\left[\tilde{V}^\dagger\Delta - \Delta^\dagger\tilde{V} + \Delta^\dagger\Delta + (\tilde{V}^\dagger)^2\Delta^2 + (\Delta^\dagger)^2\Delta^2 + 2\tilde{V}^\dagger\Delta^\dagger\Delta^2 + \tilde{V}^\dagger\tilde{V}\Delta\Delta^\dagger + \tilde{V}\Delta(\Delta^\dagger)^2\right]. \tag{59}$$

Using the expression (59) for W, $s(\tilde{V}) \leq 2$, and the properties of the functional s in Lemma 1, we obtain

$$s(W) \leq 2s(\Delta) + 9[s(\Delta)]^2 + 6[s(\Delta)]^3 + [s(\Delta)]^4. \tag{60}$$

Hence we have

$$s(V_C) \leq [1 - 2s(\Delta)]^{-r} + O\left(r\left[s(\Delta)\right]^2\right). \tag{61}$$

We can evaluate $s(\Delta)$ as

$$s(\Delta) = s(V - \tilde{V}) = \sum_{|m|>M}|J_m(z)|. \tag{62}$$

Using this expression we obtain the result (51) required for the Lemma □.

In reality we will perform a truncated operator $\tilde{V}_C(U)$ which has the higher powers of U truncated, but from the definition of s in Eq. (42), $s(\tilde{V}_C) \leq s(V_C)$. Therefore the result in Lemma 2 means that the value of r such that the sum over $|a_m|$ is approximately bounded by 2 is

$$r \approx \frac{\log 2}{2\sum_{|m|>M}|J_m(z)|}. \tag{63}$$

There is an important symmetry of the coefficients that appear in all stages of the algorithms.

Lemma 3 *At all stages in Algorithms 1 and 2 the operation that is performed corresponds to a sum of powers of U of the form $\sum_n c_n U^n$ where $c_{-n} = (-1)^n c_n$.*

Proof. This symmetry holds for $\tilde{V}$ and V due to the properties of Bessel functions [20, 10.4.1]. Then all operations that are performed can be obtained from sums and products of $\tilde{V}$ and V, as well as by truncating higher-order terms. For example, Eqs. (57) and (59) are used for V_C. Then $\tilde{V}_C$ can be obtained from V_C by omitting higher-order terms.

It is obvious that all sums of functions with this symmetry retain the symmetry. It is also obvious that omitting higher-order terms retains the symmetry. It can also be shown that products of functions with this symmetry retain the symmetry via

$$\begin{aligned}\sum_m c_m U^m \sum_k d_k U^k &= \sum_n \left(\sum_m c_m d_{n-m} \right) U^n \\ &= \sum_n \left(\sum_m (-1)^m c_{-m} (-1)^{n-m} d_{-(n-m)} \right) U^n \\ &= \sum_n (-1)^n \left(\sum_m c_{-m} d_{-(n-m)} \right) U^n \\ &= \sum_n (-1)^n \left(\sum_m c_m d_{n-m} \right) U^{-n}, \end{aligned} \tag{64}$$

where in the last line the substitution $m \mapsto -m$ and $n \mapsto -n$ has been made. Because all operations used in the algorithms are obtained from $\tilde{V}$ and V by procedures that retain the symmetry, the symmetry is retained in the algorithms $\square$.

Next we prove a lemma bounding the error due to truncating the superposition for the correction.

Lemma 4 *In Algorithm 1, the coefficients a_m required for the correction operator $V_C(U)$ satisfy*

$$\sum_{|m|>N} |a_m| \lesssim 2^{r+1} \left(\frac{z\zeta}{M} \right)^{N+1}, \tag{65}$$

where $\zeta \approx 1.8$ is the solution of $e^{1+1/2\zeta} = 2\zeta$, and N is a non-negative integer.

Proof. A trick to bound the sum over $|a_m|$ is to define a modified function V_C^+, which is the same as V_C except with the absolute values of the coefficients, and consider $V_C^+(x)$, where x is real and positive. Using Lemma 3 to give $|a_m| = |a_{-m}|$, and assuming that $x \geq 1$, we obtain

$$\begin{aligned}\sum_{|m|>N} |a_m| &= 2 \sum_{m=N+1}^{\infty} |a_m| \\ &\leq \frac{2}{x^{N+1}} \sum_{m=N+1}^{\infty} |a_m| x^m \\ &\leq \frac{2}{x^{N+1}} V_C^+(x), \end{aligned} \tag{66}$$

where $V_C^+(x) := \sum_{m=-\infty}^{\infty} |a_m| x^m$. Note that in this expression we can take N to be any integer ≥ 0. To upper bound $V_C^+(x)$ we can use the expression (57) together with (58), and

take the absolute values of all coefficients in Eq. (58). That is,

$$\begin{aligned} V_C^+(x) &\leq \sum_{k=0}^{\infty} \binom{r+k-1}{r-1} [W_+(x)]^k \\ &= [1 - W_+(x)]^{-r}, \end{aligned} \tag{67}$$

where W_+ is the function W modified to take the absolute values of all coefficients.

Using Eq. (58), we obtain

$$\begin{aligned} W_+(x) \leq \frac{1}{2} \Bigg[& \sum_{q=-\infty}^{\infty} |J_q(z)| x^{-q} \sum_{|n|>M} |J_n(z)| x^n + \sum_{|n|>M} |J_n(z)| x^{-n} \sum_{q=-M}^{M} |J_q(z)| x^q \\ & + \left(\sum_{q=-\infty}^{\infty} |J_q(z)| x^{-q} \sum_{|n|>M} |J_n(z)| x^n \right)^2 \\ & + \sum_{q=-\infty}^{\infty} |J_q(z)| x^{-q} \sum_{|n|>M} |J_n(z)| x^n \sum_{|m|>M} |J_m(z)| x^{-m} \sum_{p=-M}^{M} |J_p(z)| x^p \Bigg] \\ \leq & \sum_{q=-\infty}^{\infty} |J_q(z)| x^q \sum_{|n|>M} |J_n(z)| x^n + \left(\sum_{q=-\infty}^{\infty} |J_q(z)| x^q \sum_{|n|>M} |J_n(z)| x^n \right)^2. \end{aligned} \tag{68}$$

We will take $x = M/(z\zeta)$, where $\zeta \approx 1.8$ is the solution of $e^{1+1/2\zeta} = 2\zeta$. This number can be obtained as $1/(2\,\mathrm{ProductLog}[1/e])$ in Mathematica. It is then easy to check numerically that the right-hand side (RHS) of (68) is no greater than $1/2$.

To address this bound analytically, we can use the upper bound

$$|J_m(z)| \leq \frac{1}{|m|!} \left| \frac{z}{2} \right|^{|m|}. \tag{69}$$

Using this upper bound gives

$$\sum_{m=-\infty}^{\infty} |J_m(z)| x^m \leq e^{xz/2} + (e^{z/2x} - 1), \tag{70}$$

and

$$\sum_{|m|>M} |J_m(z)| x^m \leq 2 \frac{(xz/2)^{M+1}}{(M+1)!} + 2 \frac{(z/2x)^{M+1}}{(M+1)!}, \tag{71}$$

provided that $xz \leq M+2$ and $z/x \leq M+2$. We take $x = M/(z\zeta)$, in which case the inequalities are satisfied. For $M > 2$ the second terms in Eqs. (70) and (71) are much smaller and may be omitted. Keeping the lowest-order term in the expansion of $W_+(x)$, we obtain

$$W_+(x) \lesssim 2 \frac{(xz/2)^{M+1}}{(M+1)!} e^{xz/2}. \tag{72}$$

Using $x = M/(z\zeta)$ and Stirling's approximation, it can be shown that the RHS decreases with M, and is $< 1/2$. This part of the derivation is for $M > 2$, and for the case $M = 2$

it is easily checked numerically that the RHS is no greater than $1/2$. The choice of ζ is so that the asymptotic expression decreases with M. Hence $1 - W_+(x) \gtrsim 1/2$, so Eq. (67) gives $V_C^+(x) \lesssim 2^r$. Using this bound in Eq. (66) gives Eq. (65), as required $\square$.

If we want the correction to give no more than a factor of 2 to the complexity, we can choose $N = 3rM$, so

$$\sum_{|m|>N} |a_m| \lesssim 2^{r+1} \left(\frac{z\zeta}{M}\right)^{3rM+1}. \tag{73}$$

Since this sum of terms gives the order of the error when the correction is used, we have an error from this correction that is exponentially small, as expected. If we require higher accuracy, we could increase N so there are more terms in the correction. However, it is also possible to obtain higher accuracy by increasing M. Since that does not increase the complexity (except for a factor of 2) until $N = 3rM$, it is advantageous to simply take $N = 3rM$.

Next we can prove the overall result for the simulation with the correction.

Proof of Theorem 1. To prove this Theorem we use Algorithm 1 and select $N = 3rM$. Algorithm 1 proceeds by performing r segments as in step 2, where each segment uses a superposition of powers of the step of the quantum walk with a cutoff of M together with OAA. Then the correction in step 3 uses a sum over powers of U truncated at $N = 3rM$, with weightings a_m. Then OAA on the entire operation is used in step 4 to obtain success probability near 1.

The complexity in this Theorem is obtained from two requirements. First, we require M to be sufficiently large that $s(\tilde{V}_C)$ is not larger than 2, so we can perform the final OAA with a single step. Second, we require that the error at the end is no greater than ϵ. After the correction, the state is of the form (31), where $s = s(\tilde{V}_C)$, and

$$s(\tilde{V}_C) = \sum_{m=-N}^{N} |a_m|. \tag{74}$$

That is, there is a $1/s(\tilde{V}_C)$ factor on the amplitude for success, which is flagged by zeros in the ancilla qubits.

Using the result of Lemma 2, there is the upper bound

$$\begin{aligned} s(\tilde{V}_C) &\lesssim \left(1 - 2 \sum_{|m|>M} |J_m(z)|\right)^{-r} \\ &\approx \exp\left(2r \sum_{|m|>M} |J_m(z)|\right). \end{aligned} \tag{75}$$

Therefore we may obtain $s(\tilde{V}_C) \lesssim 2$ for

$$r \lesssim \frac{\log 2}{2\sum_{|m|>M} |J_m(z)|}. \tag{76}$$

Using the inequality

$$\frac{\log 2}{2\sum_{|m|>M} |J_m(z)|} \gtrsim \frac{(M+1)!\log 2}{8|z/2|^{M+1}}, \tag{77}$$

we find that provided

$$r \lesssim \frac{(M+1)!\log 2}{8|z/2|^{M+1}} \tag{78}$$

is satisfied, we still obtain $s(\tilde{V}_C) \lesssim 2$. This is because Eq. (78) together with Eq. (77) implies Eq. (76). In order to obtain a simulation for overall time t, we require $r = \Theta(tXd)$, where X can be taken equal to $\|H\|_{\max}$, and therefore $r = \Theta(\tau)$. According to Eq. (78), we can obtain $s(\tilde{V}_C) \lesssim 2$ with $M = \Theta(\log\tau/\log\log\tau)$.

Next we consider the criterion that the error be no larger than ϵ, which may possibly require a larger value of M. Using Lemma 4, we have

$$\sum_{|m|>N} |a_m| \lesssim 2^{r+1}\left(\frac{z\zeta}{M}\right)^{3rM+1}. \tag{79}$$

This expression gives the scaling in the error after the correction. Using Eq. (59), the error after OAA is of the same order.

Now, if the RHS is less than the allowable error ϵ with $M = \Theta(\log\tau/\log\log\tau)$, then we immediately have the error appropriately bounded with complexity $O(\tau\log\tau/\log\log\tau)$. Otherwise we can increase M, and still obtain $s(\tilde{V}_C) \lesssim 2$. Taking $3rM = O(\log(1/\epsilon))$ is sufficient to obtain error no larger than ϵ. Since $3rM$ is the same as the complexity up to a multiplicative factor, the overall result for the complexity is

$$O\left(\tau\frac{\log\tau}{\log\log\tau} + \log(1/\epsilon)\right) \tag{80}$$

□.

6 Two Rounds of Error Correction

Next we consider the result with a second round of correction, as in Algorithm 2. That is, we combine the multiple segments followed by a correction from the previous section into a compound segment; we repeat this compound segment multiple times, then perform a correction. This technique enables the scaling in the following Theorem.

Theorem 2 *A d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error $\epsilon > 0$ with*

$$O\left(\tau\frac{\log\log\tau}{\log\log\log\tau} + \log(1/\epsilon)\right) \tag{81}$$

queries, where $\tau := t\|H\|_{\max}d$.

In order to show this result, we again need to show a number of intermediate results. The first is a bound on the magnitudes of the individual coefficients a_m, rather than the sum.

Lemma 5 *The coefficients a_m for the correction operator $V_C(U)$ satisfy*

$$|a_m| \lesssim 2^r\left(\frac{z\zeta}{M}\right)^{|m|}, \tag{82}$$

where $\zeta \approx 1.8$ *is the solution of* $e^{1+1/2\zeta} = 2\zeta$.

Proof. Similar to Eq. (66), for $x \geq 1$,

$$\begin{aligned} |a_m| &\leq \sum_{n=|m|}^{\infty} |a_n| \\ &\leq \frac{1}{x^{|m|}} \sum_{n=|m|}^{\infty} |a_n| x^n \\ &\leq \frac{1}{x^{|m|}} V_C^+(x). \end{aligned} \tag{83}$$

In exactly the same way as in the proof of Lemma 4, we can take $x = M/(z\zeta)$, to obtain $W_+(x) \lesssim 1/2$ and $V_C^+(x) \leq 2^r$. We then obtain Eq. (82) as required □.

This result shows that the $|a_m|$ exponentially decrease, so we can use this to show that the sum over $|a_m|$ is properly bounded, and we can even bound a sum over $|a_m|x^m$. A key step in the proof for a single round of error correction is Eq. (71), which bounds the difference of the coefficients of the ideal sequence and the sequence we have just performed. Therefore, if we can bound that for the corrected series, then we can show that the recursion works.

The particular result we find is as in the following Lemma.

Lemma 6 *Consider Algorithm 2, with* $N = 3Mr$, *and* M *chosen such that*

$$r \leq \frac{\log 2}{2\sum_{|m|>M} |J_m(z)|}. \tag{84}$$

The coefficients a'_m *for the correction* $V'_C(U)$ *satisfy*

$$\sum_{m=-\infty}^{\infty} |a'_m| \lesssim \left\{ 1 - 2 \sum_{|m|>N} |a_m| \right\}^{-r'}. \tag{85}$$

Proof. First we define operators for what is achieved with r steps and correction for the compound segment, in a similar way as we did for the first round of correction. We use primed variables to indicate the new variable names corresponding to those for the first round of correction. The exact operation, if there were perfect correction on the compound segment, is denoted V', and the actual operation performed is denoted $\tilde{V}'$. The difference is denoted Δ', and $\Delta' = \tilde{V}' - V'$. As before these are all functions of the step operator U. These functions may be written as

$$\begin{aligned} V' &= V_C V_{\text{oaa}}^r \\ \tilde{V}' &= \tilde{V}_C V_{\text{oaa}}^r \\ \Delta' &= \left(V_C - \tilde{V}_C \right) V_{\text{oaa}}^r. \end{aligned} \tag{86}$$

Following exactly the same derivation for these primed quantities as in the proof of Lemma 2, we have

$$V'_C = \sum_{k=0}^{\infty} \binom{r'+k-1}{r'-1} W'^k, \tag{87}$$

where

$$W' = \frac{1}{2}\left(V'^{\dagger}\Delta' - \Delta'^{\dagger}\tilde{V}' + V'^{\dagger}\Delta' V'^{\dagger}\Delta' + V'^{\dagger}\Delta'\Delta'^{\dagger}\tilde{V}'\right). \tag{88}$$

We can rewrite W' as

$$W' = \frac{1}{2}(V_{\rm oaa}^{\dagger}V_{\rm oaa})^r\left(V_C^{\dagger}\Delta_C - \Delta_C^{\dagger}\tilde{V}_C\right) + \frac{1}{2}(V_{\rm oaa}^{\dagger}V_{\rm oaa})^{2r}\left(V_C^{\dagger}\Delta_C V_C^{\dagger}\Delta_C + V_C^{\dagger}\Delta_C\Delta_C^{\dagger}\tilde{V}_C\right), \tag{89}$$

where $\Delta_C := V_C - \tilde{V}_C$. Next, $V_{\rm oaa}^{\dagger}V_{\rm oaa}$ can be expanded as

$$\begin{aligned} V_{\rm oaa}^{\dagger}V_{\rm oaa} &= \left[\frac{3}{2}\tilde{V}^{\dagger} - \frac{1}{2}\tilde{V}^{\dagger}\tilde{V}\tilde{V}^{\dagger}\right]\left[\frac{3}{2}\tilde{V} - \frac{1}{2}\tilde{V}\tilde{V}^{\dagger}\tilde{V}\right] \\ &= \frac{9}{4}\tilde{V}^{\dagger}\tilde{V} - \frac{3}{2}(\tilde{V}^{\dagger}\tilde{V})^2 + \frac{1}{4}(\tilde{V}^{\dagger}\tilde{V})^3 \\ &= \mathbb{1} - \frac{3}{2}\Delta^{\dagger}\Delta + \frac{3}{4}(\Delta^{\dagger}\Delta)^2 - \frac{1}{4}(\Delta^{\dagger}\Delta)^3 - \frac{3}{4}\left((\tilde{V}^{\dagger}\Delta)^2 + (\Delta^{\dagger}\tilde{V})^2\right) \\ &\quad - \frac{1}{4}\left((\tilde{V}^{\dagger}\Delta)^3 + (\Delta^{\dagger}\tilde{V})^3\right) - \frac{3}{4}\Delta^{\dagger}\Delta\left(\Delta^{\dagger}\tilde{V} + \tilde{V}^{\dagger}\Delta\right). \end{aligned} \tag{90}$$

Therefore, using the properties of the functional s and $s(\tilde{V}) \leq 2$,

$$\begin{aligned} s(V_{\rm oaa}^{\dagger}V_{\rm oaa}) &\leq 1 + \frac{3}{2}[s(\Delta)]^2 + \frac{3}{4}[s(\Delta)]^4 + \frac{1}{4}[s(\Delta)]^6 + 6[s(\Delta)]^2 + 4[s(\Delta)]^3 + 3[s(\Delta)]^3 \\ &= 1 + \frac{15}{2}[s(\Delta)]^2 + 7[s(\Delta)]^3 + \frac{3}{4}[s(\Delta)]^4 + \frac{1}{4}[s(\Delta)]^6. \end{aligned} \tag{91}$$

Given that we have chosen M so that Eq. (63) holds, we have

$$\begin{aligned} s\left((V_{\rm oaa}^{\dagger}V_{\rm oaa})^r\right) &\leq \left(1 + \frac{15}{2}[s(\Delta)]^2 + O\left([s(\Delta)]^3\right)\right)^{\frac{\log 2}{2s(\Delta)}} \\ &= 1 + \frac{15}{4}(\log 2)s(\Delta) + O\left([s(\Delta)]^2\right). \end{aligned} \tag{92}$$

In addition, the restriction (84) ensures that $s(V_C) \leq 2$. Hence we have

$$s(W') \leq 2s(\Delta_C) + O\left(s(\Delta)s(\Delta_C)\right). \tag{93}$$

Using Eq. (87) we then have

$$s(V_C') \lesssim \left[1 - 2s(\Delta_C)\right]^{-r'}. \tag{94}$$

Recognising that $s(\Delta_C) \leq \sum_{|m|>N}|a_m|$, this gives the result required □.

Using Lemma 4, and taking $N = 3rM$, we then find that we can take r' satisfying

$$r' \lesssim \frac{\log 2}{2^{r+2}}\left(\frac{M}{z\zeta}\right)^{3rM+1}, \tag{95}$$

and obtain $s(V_C') \leq 2$. This expression is the equivalent of Eq. (78). Next we prove a lemma bounding the size of the error.

Lemma 7 *Consider Algorithm 2, with $N = 3Mr$, and M chosen such that*

$$r \leq \frac{\log 2}{2\sum_{|m|>M} |J_m(z)|}. \tag{96}$$

The coefficients a'_m for the correction $V'_C(U)$ satisfy

$$\sum_{|m|>N'} |a'_m| \lesssim 2^{r'} \left(\frac{z\zeta\zeta' 2^{1/M}}{M} \right)^{N'+1}, \tag{97}$$

where $\zeta \approx 1.8$ is the solution of $e^{1+1/2\zeta} = 2\zeta$ and $\zeta' \approx 1.5$ is a solution of $\zeta'^5(\sqrt{2} - 2\zeta')^2 = 16\sqrt{2}$.

Proof. In the same way as for Lemma 4, we have

$$\sum_{|m|>N'} |a'_m| \leq \frac{2}{x^{N'+1}} V_C^{+\prime}(x), \tag{98}$$

where $V_C^{+\prime}(x) := \sum_{m=-\infty}^{\infty} |a'_m| x^m$. This function satisfies

$$V_C^{+\prime}(x) \leq \left[1 - W'_+(x)\right]^{-r'}, \tag{99}$$

where W'_+ is the function W' modified to take the absolute values of all coefficients.

To bound $W'_+(x)$, we can use Eq. (89) and Eq. (90). For $V_{\text{oaa}}^\dagger V_{\text{oaa}}$ with the absolute values of all coefficients taken, it will be upper bounded by

$$1 + \frac{3}{2}\delta^2 + \frac{3}{4}\delta^4 + \frac{1}{4}\delta^6 + \frac{3}{2}\nu^2\delta^2 + \frac{1}{2}\nu^3\delta^3 + \frac{3}{2}\nu\delta^3, \tag{100}$$

where

$$\nu := \sum_{q=-M}^{M} |J_q(z)| x^q, \tag{101}$$

$$\delta := \sum_{|n|>M} |J_n(z)| x^n. \tag{102}$$

These expressions are upper bounded in Eqs. (70) and (71). It was found that for $x \leq M/(z\zeta)$, $\nu\delta < 1/2$. In addition, for this choice of x, $\nu \gg 1$ and $\delta \ll 1$, so the overall value will be no greater than 2. Then, using (89), we obtain

$$W'_+(x) \leq 2^r \sum_{q=-\infty}^{\infty} |a_q| x^q \sum_{|n|>N} |a_n| x^n + \left(2^r \sum_{q=-\infty}^{\infty} |a_q| x^q \sum_{|n|>N} |a_n| x^n \right)^2. \tag{103}$$

Note that we can use $|a_m| = |a_{-m}|$ due to symmetry. Then, using the bound (82), we have

$$\sum_{|n|>N} |a_n| x^n \leq \frac{2^{r+1}}{1 - z\zeta x/M} \left(\frac{z\zeta x}{M} \right)^{N+1}. \tag{104}$$

Similarly the sum over all powers can be bounded as

$$\sum_{q=-\infty}^{\infty} |a_q| x^q \leq \frac{2^{r+1}}{1 - z\zeta x/M}. \tag{105}$$

Therefore we have

$$\sum_{|m|>N'} |a'_m| \lesssim \frac{1}{x^{N'+1}} \left\{ 1 - 4 \frac{2^{3r}}{(1 - z\zeta x/M)^2} \left(\frac{z\zeta x}{M} \right)^{N+1} \right\}^{-r'}. \tag{106}$$

We now wish to take x to be slightly less than $M/(z\zeta 2^{1/M})$, so that the expression in braces is $\geq 1/2$. In particular we take $x = M/(z\zeta\zeta' 2^{1/M})$, where $\zeta' \approx 1.52937$ is a solution of $\zeta'^5(\sqrt{2} - 2\zeta')^2 = 16\sqrt{2}$. Then we obtain

$$\sum_{|m|>N'} |a'_m| \lesssim \left(\frac{z\zeta\zeta' 2^{1/M}}{M} \right)^{N'+1} 2^{r'} \tag{107}$$

□.

We are now in a position to prove the Theorem for the complexity.

Proof of Theorem 2. For this proof we use Algorithm 2. The simulation proceeds by using compound segments, where each segment uses r segments and a correction. We perform r' of these compound segments, followed by an overall correction. Now the overall length of the simulation is rr', so we require $rr' = \Theta(\tau)$. We have three requirements:

1. The corrections for the compound segments satisfy $s(V_C) \leq 2$, so OAA can be performed in one step.
2. The final correction satisfies $s(V'_C) \leq 2$, so the final OAA can be performed in one step.
3. The error as obtained in Lemma 7 is upper bounded by ϵ.

Considering the first requirement, let us take

$$M = \Theta\left(\frac{\log\log\tau}{\log\log\log\tau} \right). \tag{108}$$

Then we have Eq. (63) satisfied for $r = \Theta(\log\tau)$, which implies that $s(V_C) \leq 2$. Second, we find that we have Eq. (95) satisfied with $r' = \Theta(\tau/r)$. Then we obtain $s(V'_C) \leq 2$ for the final OAA, and $rr' = \Theta(\tau)$ as required. Because the complexity of step 2 of Algorithm 2 is $rr'M$ up to multiplying factors, it gives a contribution to the complexity of

$$\Theta\left(\tau \frac{\log\log\tau}{\log\log\log\tau} \right). \tag{109}$$

Finally we consider the third requirement. We can choose $N' = 9rr'M$ without changing the complexity. Now if the expression on the RHS of Eq. (97) in Lemma 7 is less than ϵ, then we have satisfied this requirement. Otherwise we can further increase M in order to obtain error no greater than ϵ. The overall complexity is equal to N' up to multiplying factors. We can obtain the RHS of Eq. (97) less than ϵ by taking $N' = O(\log(1/\epsilon))$. Hence the overall complexity sufficient to satisfy both requirements is as given in Eq. (81) □.

7 Conclusions

We have shown how to perform corrections on the superposition of quantum walk steps approach to Hamiltonian simulation from Ref. [16]. This approach gives a result much closer to the lower bound for complexity than Ref. [16], because it has a sum rather than a product in the scaling of the complexity. Our result is very close to the lower bound for the complexity in Ref. [16], except our result differs by double-logarithmic factors for the scaling in τ and ϵ.

Our approach to correcting the quantum walk is sufficiently flexible that it can be used to perform an arbitrary number of rounds of correction. That should provide scaling of the complexity with further iterated logarithms of τ, though not strictly linear scaling in τ. Proving the complexity scaling is quite complicated, so we have limited to analyzing two rounds of correction here.

After completion of this work, we were made aware of another approach that yields complexity closer to the lower bound [19]. On the other hand, our result is more flexible than that in Ref. [19], because it can be applied to simulations with sums of many different unitary operators. In contrast, the method of Ref. [19] only applies to using a single unitary operator to simulate Hamiltonian evolution. This means that our approach can be applied to both simulations based on quantum walks, and simulations based on a Taylor series [16], whereas the method of Ref. [19] does not apply to Taylor series. That is another important case, because it seems more useful for quantum chemistry, and also has better scaling in the number of additional gates. In addition, it is needed for Hamiltonians that are a sum of operators sparse in different bases, which the quantum walk approach cannot be applied to. For example, particles in a potential have a Hamiltonian of this type [21].

Acknowledgements

We thank Aaron Ostrander, Aram Harrow and Andrew Childs for valuable discussions. We acknowledge support from IARPA contract number D15PC00242. DWB is funded by an Australian Research Council Future Fellowship (FT100100761) and a Discovery Project (DP160102426). LN thanks the support from Fundação para a Ciência e a Tecnologia (Portugal), namely through programmes PTDC/POPH/POCH and projects UID/EEA/50008/2013, IT/QuSim, ProQuNet, partially funded by EU FEDER, and from the EU FP7 project PAPETS (GA 323901). Furthermore, LN acknowledges the support from the DP-PMI and FCT (Portugal) through scholarship SFRH/BD/52241/2013.

References

1. R. P. Feynman (1982), *Simulating physics with computers*, International Journal of Theoretical Physics, 21, pp. 467-488.
2. S. Lloyd (1996), *Universal quantum simulators*, Science, 273, pp. 1073-1078.
3. D. Aharonov and A. Ta-Shma (2003), *Adiabatic quantum state generation and statistical zero knowledge*, in Proceedings of the 35th ACM Symposium on Theory of Computing, pp. 20-29.
4. A. W. Harrow, A. Hassidim, and S. Lloyd (2009), *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett., 103, p. 150502.
5. A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman (2003), *Exponential algorithmic speedup by quantum walk*, in Proceedings of the 35th ACM Symposium on Theory of Computing, pp. 59-68.

6. A. M. Childs, R. Cleve, S. P. Jordan, and D. Yonge-Mallo (2009), *Discrete-query quantum algorithm for NAND trees*, Theory of Computing, 5, pp. 119-123.
7. A. M. Childs, R. Kothari, and R. D. Somma (2015), *Quantum linear systems algorithm with exponentially improved dependence on precision*, arXiv:1511.02306.
8. D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders (2007), *Efficient quantum algorithms for simulating sparse Hamiltonians*, Communications in Mathematical Physics, 270, pp. 359-371.
9. N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders (2011), *Simulating quantum dynamics on a quantum computer*, Journal of Physics A: Mathematical and Theoretical, 44, p. 445308.
10. A. M. Childs (2010), *On the relationship between continuous- and discrete-time quantum walk*, Communications in Mathematical Physics, 294, pp. 581-603.
11. D. Poulin, A. Qarry, R. D. Somma, and F. Verstraete (2011), *Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space*, Phys. Rev. Lett., 106, p. 170501.
12. D. W. Berry and A. M. Childs (2012), *Black-box Hamiltonian simulation and unitary implementation*, Quantum Inf. Comput., 12, pp. 29-62.
13. A. M. Childs and N. Wiebe (2012), *Hamiltonian simulation using linear combinations of unitary operations*, Quantum Inf. Comput., 12, pp. 901-924.
14. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma (2014), *Exponential improvement in precision for simulating sparse Hamiltonians*, in Proceedings of the 46th Annual ACM Symposium on Theory of Computing, pp. 283-292.
15. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma (2015), *Simulating Hamiltonian dynamics with a truncated Taylor series*, Phys. Rev. Lett., 114, p. 090502.
16. D. W. Berry, A. M. Childs, and R. Kothari (2015), *Hamiltonian simulation with nearly optimal dependence on all parameters*, in Proceedings of the 56th IEEE Symposium on Foundations of Computer Science, pp. 792-809.
17. R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik (2016), *Exponentially more precise quantum simulation of fermions in second quantization*, New Journal of Physics, 18, p. 033032.
18. R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik (2015), *Exponentially more precise quantum simulation of fermions II: Quantum chemistry in the CI matrix representation*, arXiv:1506.01029.
19. G. H. Low and I. L. Chuang (2016), *Optimal Hamiltonian simulation by quantum signal processing*, arXiv:1606.02685.
20. F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark (2010), *NIST Handbook of Mathematical Functions*, Cambridge University Press (New York, NY).
21. R. D. Somma (2016), *Quantum simulations of one dimensional quantum systems*, Quantum Inf. Comput., 16, pp. 1125-1168.