Exact and efficient Lanczos method on a quantum computer

William Kirby[1,2], Mario Motta[3], and Antonio Mezzacapo[4]

[1] IBM Quantum, IBM Research Cambridge, Cambridge, MA 02142, USA

[2] Department of Physics and Astronomy, Tufts University, Medford, MA 02155, USA

[3] IBM Quantum, IBM Research Almaden, San Jose, CA 95120, USA

[4] IBM Quantum, T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

We present an algorithm that uses block encoding on a quantum computer to exactly construct a Krylov space, which can be used as the basis for the Lanczos method to estimate extremal eigenvalues of Hamiltonians. While the classical Lanczos method has exponential cost in the system size to represent the Krylov states for quantum systems, our efficient quantum algorithm achieves this in polynomial time and memory. The construction presented is exact in the sense that the resulting Krylov space is identical to that of the Lanczos method, so the only approximation with respect to the exact method is due to finite sample noise. This is possible because, unlike previous quantum Krylov methods, our algorithm does not require simulating real or imaginary time evolution. We provide an explicit error bound for the resulting ground state energy estimate in the presence of noise. For our method to be successful efficiently, the only requirement on the input problem is that the overlap of the initial state with the true ground state must be $\Omega(1/\textbf{poly}(n))$ for n qubits.

1 Introduction

Estimating the ground state energy of a quantum system has broad applications across physics and chemistry. In the most challenging cases, classically accomplishing this is believed to be exponentially hard in the system size, due to the cost of classically representing vectors in the Hilbert space [1, 2, 3, 4,

William Kirby: william.kirby@ibm.com

5, 6]. Many quantum algorithms for the task have been proposed (see for example [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]) that avoid this representation problem by encoding the system's Hilbert space in the Hilbert space of a register of qubits, whose dimension also grows exponentially.

The problem of finding ground state energies of local Hamiltonians is QMA-hard [30, 31]. This means that any sufficiently general quantum algorithms for approximating ground state energies must require additional assumptions. Some typical additional assumptions are finding an initial guess state that has $\Omega(1/\text{poly}(n))$ overlap with sufficiently low-lying energy states (for n qubits) [7, 20, 28, 29], finding a gapped adiabatic path from an efficiently solvable Hamiltonian to the target Hamiltonian [9], optimizing a variational quantum circuit [2], or assuming bounded correlation length in the ground state [16].

One design principle for quantum algorithms is to reduce the overhead of classical numerical techniques. This can be applied to classical algorithms that target ground states. A well-known classical technique for finding extremal eigenvalues is the *Lanczos method* [32, 33], which constructs a linear *Krylov space* as the span of states obtained by applying powers of the Hamiltonian to some reference state. The Hamiltonian is then diagonalized within the Krylov space, obtaining a variational estimate of the ground state energy. This method has been widely studied and applied, and for a given problem instance it converges exponentially quickly in the dimension of the Krylov space, in practice often reaching sufficient accuracy within only some tens of powers of the Hamiltonian [34, 35, 36, 37]. However, for quantum systems, the Krylov basis

vectors have exponential dimension in the system size, which makes the classical Lanczos method exponentially costly in general.

A natural question is therefore whether a quantum version of the algorithm can make this cost polynomial. Applying a power of the Hamiltonian operator to a reference state is a nonunitary operation. Hence, some quantum versions of the Lanczos method have been proposed that replace powers of the Hamiltonian with unitary operators that approximate real [17, 18, 21, 25, 23] or imaginary [16] time evolutions generated by the Hamiltonian. Another strategy approximates powers of the Hamiltonian itself using linear combinations of time evolutions [24]. These methods avoid the exponential cost of representing quantum states, which have dimension 2^n for n qubits. However, they only approximate the original Lanczos method due to the modification of the Krylov basis vectors. Furthermore, real and imaginary time evolutions can only be approximated on a digital quantum computer.

In this paper, we propose a quantum algorithm that exactly reproduces the Krylov space used in the classical Lanczos method up to finite sample noise, without the exponential classical representation cost, thus avoiding the aforementioned approximation errors. Unlike in typical classical implementations, our method does not involve iteratively orthogonalizing the Krylov vectors, but it does yield the same Krylov space. The algorithm we propose is based on the observation that the simplest case of qubitization presented in [38] applies to a Hamiltonian expressed as a linear combination of Pauli operators and produces block encodings of Chebyshev polynomials of the Hamiltonian. It does not require quantum signal processing [39], which would be required to approximate time evolution using qubitization. Instead, we show that applications of the block-encoding unitaries followed by measurement, repeated in order to estimate a collection of expectation values, provide enough information for the classical computer to do the rest.

We also provide an explicit analysis of the ground state energy error from our algorithm in the presence of noise. The resulting Krylov space dimension (which is proportional to the maximum circuit depth) required to achieve a given target error scales either as the inverse of the target error or of the spectral gap, whichever is more favorable, times logarithmic factors: see (38). This means that our method can tolerate a vanishing spectral gap, but also that when the spectral gap is nonzero, the total number of measurements required to achieve a given target error scales asymptotically as the inverse-squared error (times logarithmic factors): see (44). This is the scaling one would expect in an algorithm that is based on repeated sampling.

Compared to prior work, if we first compare to previous quantum subspace diagonalization methods [16, 17, 18, 21, 24, 25, 23], unlike all of these our algorithm is exactly equivalent to the classical Lanczos method up to sampling noise, and does not require approximate real or imaginary time evolution. Unlike imaginary time-evolution, our algorithm does not require finite correlation length [16]. Compared to adiabatic state preparation [9], our algorithm does not depend on the spectral gap unless that is larger than the target error, and unlike variational quantum eigensolvers [2], it does not require optimizing a parameterized quantum circuit. Finally, compared to quantum phase estimation, our algorithm requires a number of shots that is inverse quadratic in the target error rather than merely inverse (which is preferable). However, it has shorter circuits if the phase estimation algorithm is also based on block encoding, and our method is robust to noise. Details of all of these comparisons are given in Section 6.

The paper is organized as follows. In Section 2 we give requisite background on block encoding, qubitization, and quantum subspace diagonalization. In Section 3, we describe and analyze our quantum algorithm for producing a Krylov space. In Section 4, we describe the classical post-processing (regularization) required for our method, provide an error bound for the resulting ground state energy estimate in the presence of noise, and discuss the resulting scalings of Krylov space dimension and measurements. In Section 5, we provide some numerical demonstrations of our method. Finally, we discuss our results and conclude in Section 6.

2 Background

2.1 Block encoding

Definition 1 (block encoding [38]). A *block encoding* of a Hamiltonian H (acting on a Hilbert space $\mathcal{H}_s$ whose states are denoted $|\cdot\rangle_s$) is a pair (U, G), where U is a unitary acting on $\mathcal{H}_a \otimes \mathcal{H}_s$ (for some auxiliary Hilbert space $\mathcal{H}_a$ whose states are denoted $|\cdot\rangle_a$), and $|G\rangle_a := G|0\rangle_a$ is a state of the auxiliary qubits such that

$$(\langle G|_a \otimes \mathbb{1}_s)U(|G\rangle_a \otimes \mathbb{1}_s) = H. \tag{1}$$

Here $\mathbb{1}_s$ denotes identity acting on $\mathcal{H}_s$.

As an example of block encoding, consider an n-qubit Hamiltonian expressed as a linear combination of Pauli operators:

$$H = \sum_{i=0}^{T-1} \alpha_i P_i, \tag{2}$$

where α_i are real coefficients, P_i are Pauli operators, and the number of terms is $T = O(\text{poly}(n))$. We assume that the coefficients α_i are nonnegative, and instead each Pauli P_i carries a ± 1 sign. We also require a normalization condition on H: $\sum_{i=0}^{T-1} |\alpha_i| = \sum_{i=0}^{T-1} \alpha_i = 1$. Given some non-normalized input Hamiltonian whose coefficients are $\alpha_i^{(E)}$ (units of energy), we obtain normalized coefficients as $\alpha_i = |\alpha_i^{(E)}| / \sum_{i=0}^{T-1} |\alpha_i^{(E)}|$.

The block encoding unitary is

$$U = \sum_{i=0}^{T-1} |i\rangle_a \langle i|_a \otimes P_i, \tag{3}$$

i.e., application of each Pauli term P_i controlled on the state of the auxiliary register being $|i\rangle_a$. The corresponding block encoding state is

$$|G\rangle_a = \sum_{i=0}^{T-1} \sqrt{\alpha_i}|i\rangle_a. \tag{4}$$

Inserting U as defined in (3) and $|G\rangle_a$ as defined in (4) into the left-hand side of (1) and using the fact that $\alpha_i \geq 0$ for all i, one can verify that (1) is satisfied, i.e., this (U, G) forms a valid block encoding of H.

One option for encoding $|i\rangle_a$ is to use $\lceil \log_2 T \rceil$ auxiliary qubits and let each $|i\rangle_a$ be the computational basis state corresponding to the binary number i [38]. In this case, to implement U we must, for each of the T P_i's, apply $P_i^{(j)}$ (the jth single-qubit Pauli operator in P_i) to system qubit j, controlled on the auxiliary qubits being in state $|i\rangle_a$. P_i is an n-qubit Pauli operator, so implementing U requires applying at most nT single-qubit Pauli operators, each controlled on all of the auxiliary qubits.

To prepare $|G\rangle_a = G|0\rangle_a$, we can use existing state preparation procedures, since there are only logarithmically-many auxiliary qubits and thus preparing an arbitrary state on them is efficient. For example, one can use the state preparation based on binary data structures of [40, Theorem A.1]. To prepare an arbitrary real-amplitude state of $\lceil \log_2 T \rceil$ qubits using this method requires fewer than $2T$ single-qubit rotations, each of which is controlled on up to all of the remaining qubits.

The qubitization iteration step that we will discuss below in Section 2.2 also requires implementing R, a reflection around $|G\rangle_a$. One can do this by applying $G^\dagger$, the inverse of the state preparation unitary, then applying the $\lceil \log_2 T \rceil$-controlled phase that reflects around $|0\rangle_a$, and finally reapplying G. Hence the total cost for R is at most $4T \lceil \log_2 T \rceil$-controlled gates. Methods for implementing the block encoding itself are not the main focus of this paper, but in Appendix C we present an alternative encoding based on the binary representation of the Pauli operators, which uses more qubits in exchange for shorter circuits and is well-adapted to local Pauli Hamiltonians such as spin models. When this block encoding is applied to a Heisenberg model containing arbitrary XX and ZZ interactions, U requires $3n + T$ two-qubit gates, G requires $4n^2 - 10n$ two-qubit gates plus 2 single-qubit gates, and R requires $8n^2 + 14$ two-qubit gates plus 4 single-qubit gates. Various other block encodings have been proposed that may be advantageous depending on the Hamiltonian and the available quantum device [38, 41, 42, 43].

2.2 Qubitization

The other main component we will need is the simplest version of qubitization introduced in [38],

which applies when the block encoding unitary is self-inverse, i.e.,

$$U^2 = \mathbb{1}. \tag{5}$$

In the example above, U is the product of controlled Pauli operators given by (3), so (5) is satisfied. The following lemma is based directly on Section IV in [38]; for completeness, we give its proof in Appendix A.

Lemma 1 (Chebyshev polynomials from block encoding)**.** *Given a block encoding* (U, G) *of a Hamiltonian* H, *such that* $U^2 = \mathbb{1}$, *let*

$$R := (2|G\rangle_a\langle G|_a - \mathbb{1}_a) \otimes \mathbb{1}_s \tag{6}$$

be the reflection around $|G\rangle_a$ *in the auxiliary space. Then*

$$(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes \mathbb{1}_s) = T_k(H) \tag{7}$$

for any $k = 0, 1, 2, ...,$ *where* $T_k(\cdot)$ *is the kth Chebyshev polynomial of the first kind. In other words,* $(RU)^k$ *is a block encoding of* $T_k(H)$.

2.3 Quantum subspace diagonalization

Quantum subspace diagonalization [13, 14, 17, 16, 19, 15, 18, 21, 22, 37, 24, 25, 23, 26, 27] is a method for obtaining a variational estimate of the ground state energy of a Hamiltonian H. In the most general setting, we assume access to a set of states

$$\{|\psi_k\rangle = U_k|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\} \tag{8}$$

that can be prepared on a quantum computer by quantum circuits U_k. First, we form the $D \times D$ matrices **H** and **S** whose entries are

$$\begin{aligned} \mathbf{H}_{ij} &= \langle\psi_i|H|\psi_j\rangle = \langle\psi_0|U_i^\dagger H U_j|\psi_0\rangle, \\ \mathbf{S}_{ij} &= \langle\psi_i|\psi_j\rangle = \langle\psi_0|U_i^\dagger U_j|\psi_0\rangle. \end{aligned} \tag{9}$$

S is the *overlap* (Gram) matrix of the Krylov states (8). Both can be estimated by repeated swap [44] or Hadamard tests [45]. Then, having estimated **H** and **S**, we classically solve the generalized eigenvalue problem

$$\mathbf{H}\vec{v} = \epsilon\mathbf{S}\vec{v} \tag{10}$$

and find the lowest eigenvalue ϵ. This is our variational estimate of the ground state energy, and it corresponds to the lowest expected energy of any state in the subspace

$$\text{span}\{|\psi_k\rangle = U_k|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\}. \tag{11}$$

Note that typically the generalized eigenvalue problem (10) must be regularized due to ill-conditioning of **S** as D increases [23, 37]: this is discussed in detail in Section 4.

The Lanczos method [32, 33] is a classical matrix diagonalization method that is closely related to quantum subspace diagonalization. In it, Krylov states $|\psi_i\rangle$ are obtained by multiplying some reference state $|\psi_0\rangle$ by powers of the Hamiltonian,

$$\{|\psi_k\rangle = H^k|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\}, \tag{12}$$

and these are used to construct the matrices **H** and **S**. This yields a ground state energy estimate that converges exponentially in D [36], assuming infinite precision arithmetic; the impact of noisy **H** and **S** is discussed in Section 4. In general, however, applying this method to quantum systems is exponentially costly in the number of qubits, since the Krylov states have exponential dimension.

To avoid the classical barrier of exponential dimensionality, as noted above, several methods have been proposed to permit a quantum variant of the algorithm. These include *quantum Lanczos* (QLanczos) [16], in which the powers of H are replaced by imaginary time evolutions under H, methods in which the powers of H are replaced by real time evolutions under H [17, 18, 21, 37, 25, 23], and a method for approximating powers of H using linear combinations of time evolutions [24]. All of these methods converge to the Lanczos method in appropriate limits. However, either real or imaginary time evolutions are required, both of which entail approximations.

3 Lanczos method on a quantum computer

3.1 Algorithm description

We propose a quantum algorithm that exactly produces the Krylov space used in the classical Lanczos method up to finite sample noise. We will focus on the case where the input block encoding is of

a Hamiltonian encoded as a linear combination of Pauli operators as in the example in Definition 1, since in this case the measurement scheme is particularly simple. However, the method can be generalized to other block encodings, as we will describe.

The idea is to use as the Krylov basis vectors

$$\{|\psi_k\rangle = T_k(H)|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\}. \quad (13)$$

The Chebyshev polynomials $T_k(x)$ for $k = 0, ..., D-1$ are a basis for the polynomials of degree less than D, so the Krylov space spanned by (13) is identical to the Krylov space obtained from powers of the Hamiltonian:

$$\begin{aligned}&\text{span}\{T_k(H)|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\}\\ &= \text{span}\{H^k|\psi_0\rangle \mid k = 0, 1, 2, ..., D-1\}.\end{aligned} \quad (14)$$

Since the quantum subspace diagonalization method classically finds the state with lowest expected energy in this span, the performance of the Krylov space spanned by Chebyshev polynomials of the Hamiltonian applied to the initial state will be identical to that of powers of the Hamiltonian applied to the initial state, up to noise.

Also, we reviewed in Section 2.2 (Lemma 1) how to implement block encodings of Chebyshev polynomials of a block encoded Hamiltonian. There was no approximation in Lemma 1, so as long as the input block encoding is exact, the method will exactly produce the subspace (14).

It remains to show how to extract the matrices **H** and **S** from the block encodings of $T_k(H)|\psi_0\rangle$. For **S** we have

$$\mathbf{S}_{ij} = \langle\psi_0|T_i(H)T_j(H)|\psi_0\rangle = \langle T_i(H)T_j(H)\rangle_0, \quad (15)$$

where $\langle\cdot\rangle_0$ denotes expectation value with respect to the initial state $|\psi_0\rangle$. Chebyshev polynomials satisfy the identity

$$T_i(x)T_j(x) = \frac{1}{2}\Big(T_{i+j}(x) + T_{|i-j|}(x)\Big). \quad (16)$$

Inserting this into (15) yields

$$\begin{aligned}\mathbf{S}_{ij} &= \frac{1}{2}\langle T_{i+j}(H) + T_{|i-j|}(H)\rangle_0\\ &= \frac{1}{2}\Big(\langle T_{i+j}(H)\rangle_0 + \langle T_{|i-j|}(H)\rangle_0\Big).\end{aligned} \quad (17)$$

For **H** we have

$$\mathbf{H}_{ij} = \langle T_i(H)HT_j(H)\rangle_0. \quad (18)$$

Using the fact that $H = T_1(H)$ and applying (16) twice, we obtain

$$\begin{aligned}\mathbf{H}_{ij} = \frac{1}{4}\Big(&\langle T_{i+j+1}(H)\rangle_0 + \langle T_{|i+j-1|}(H)\rangle_0\\ &+ \langle T_{|i-j+1|}(H)\rangle_0 + \langle T_{|i-j-1|}(H)\rangle_0\Big).\end{aligned} \quad (19)$$

Since $i, j = 0, 1, 2, ..., D-1$, all matrix elements in both matrices are linear combinations of the expectation values

$$\langle T_k(H)\rangle_0, \quad k = 0, 1, 2, ..., 2D-1, \quad (20)$$

where the highest value $2D-1$ comes from the first term in (19) when $i = j = D-1$. Therefore, in order to construct the matrices **H** and **S** it is enough to estimate all of the expectation values (20).

Given our block encoding (U, G) of H, Lemma 1 yielded (7), which when we take the expectation value with respect to $|\psi_0\rangle$ in the system space becomes

$$\langle T_k(H)\rangle_0 = (\langle G|_a \otimes \langle\psi_0|)(RU)^k(|G\rangle_a \otimes |\psi_0\rangle). \quad (21)$$

Since R is a reflection about $|G\rangle_a$, it is Hermitian and also acts as identity on $|G\rangle_a$, so the leftmost R in (21) can be removed, yielding

$$\langle T_k(H)\rangle_0 = (\langle G|_a \otimes \langle\psi_0|)U(RU)^{k-1}(|G\rangle_a \otimes |\psi_0\rangle). \quad (22)$$

The product of operators in (22) can be rewritten as

$$U(RU)^{k-1} = \begin{cases}(UR)^{k/2}R(RU)^{k/2} & \text{if } k \text{ is even,}\\ (UR)^{\lfloor k/2\rfloor}U(RU)^{\lfloor k/2\rfloor} & \text{if } k \text{ is odd.}\end{cases} \quad (23)$$

Therefore, if we define the state

$$|\psi_{\lfloor k/2\rfloor}\rangle = (RU)^{\lfloor k/2\rfloor}(|G\rangle_a \otimes |\psi_0\rangle), \quad (24)$$

whose adjoint is

$$\langle\psi_{\lfloor k/2\rfloor}| = (\langle G|_a \otimes \langle\psi_0|)(UR)^{\lfloor k/2\rfloor}, \quad (25)$$

then because U and R are both Hermitian, we can rewrite (22) as

$$\langle T_k(H)\rangle_0 = \begin{cases} \langle\psi_{\lfloor k/2\rfloor}|R|\psi_{\lfloor k/2\rfloor}\rangle & \text{if } k \text{ is even,} \\ \langle\psi_{\lfloor k/2\rfloor}|U|\psi_{\lfloor k/2\rfloor}\rangle & \text{if } k \text{ is odd.} \end{cases}$$
(26)

Hence, we can estimate all the matrix elements of $\mathbf{H}$ and $\mathbf{S}$ by estimating the expectation values in (26) for each $k = 0, 1, 2, ..., 2D-1$. We accomplish this as follows:

1. Prepare $|\psi_{\lfloor k/2\rfloor}\rangle$ by applying RU $\lfloor k/2\rfloor$ times to $|G\rangle_a \otimes |\psi_0\rangle$.

2. If k is even, we want to measure R. To do this, first apply $G^\dagger$ (recall that $|G\rangle_a = G|0\rangle_a$; see Definition 1). Then measure $2|0\rangle_a\langle0|_a - \mathbb{1}$ on the auxiliary qubits, i.e., measure all auxiliary qubits in the computational basis and return $+1$ if all outcomes are $|0\rangle$, otherwise -1.

3. If k is odd, we want to estimate the expectation value of

$$U = \sum_{i=0}^{T-1} |i\rangle_a\langle i|_a \otimes P_i.$$
(27)

We do this by separately estimating the expectation value of each term, so for each state preparation we measure some $|i\rangle_a\langle i|_a \otimes P_i$. Measure the auxiliary qubits in the computational basis and the system qubits in a local Pauli basis compatible with P_i. If the outcome of this measurement is $|i\rangle_a$, return the system qubits' measurement outcome for P_i, and otherwise return 0.

4. Return to step 1 and repeat until enough measurements are obtained to estimate the desired expectation value to the desired precision.

Note that in step 3 above, locally-compatible Pauli measurements can be grouped using any of the same methods as for standard Pauli Hamiltonian expectation value estimation [46, 47, 48, 49, 50, 51, 52], by measuring the system qubits in the shared local Pauli basis. Also note that by employing one of these methods, the number of shots depends on the $l1$-norm of the Pauli coefficients in
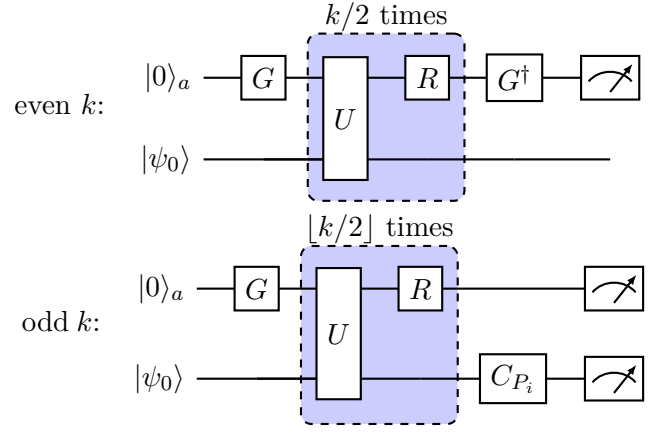

Figure 1: The quantum circuits used in our algorithm. For each $k = 0, 1, 2, ..., 2D-1$ we repeatedly sample from one of the above circuits, where D is the desired Krylov space dimension. When k is even, we apply the upper circuit. When k is odd, we apply the lower circuit for each Pauli term P_i in the Hamiltonian. All measurements are in the computational basis, and C_{P_i} is a single layer of single-qubit Clifford gates that maps P_i to a diagonal Pauli operator. As defined in the text, U block encodes the Hamiltonian H, G prepares the state that identifies the block containing H, and R reflects about that state.

the Hamiltonian, which in our case is fixed to 1 (see Definition 1), so the requirement of measuring in distinct Pauli bases does not contribute additional measurement overhead asymptotically. The above algorithm is displayed in Fig. 1.

3.2 Runtime and qubit cost

Let us summarize the requirements and costs for our quantum implementation of the Lanczos method:

1. The required input is a block encoding (U, G) of H: there are various options for constructing this, as discussed in Section 2.1. Using the example in Section 2.1, for n qubits and T Hamiltonian terms the cost of U is nT, the cost of G is $2T$, and the cost of R is $4T$, all in $\lceil\log_2 T\rceil$-controlled single-qubit gates.

2. The number of measurements depends on the desired error and the classical method for regularizing and solving the generalized eigenvalue problem (10). This analysis is provided in Section 4.

3. Each state preparation [given in (24)] requires preparing $|G\rangle_a \otimes |\psi_0\rangle$, followed by at most $D-1$ applications of RU. We then either measure in a local Pauli basis, or apply $G^\dagger$ and then measure. The latter leads to the longest coherent sequence of operations required by the algorithm. Using the costs of the block encoding in point 1 above, we find that this sequence requires at most $(D-1)nT + 4DT$ $\lceil \log_2 T \rceil$-controlled single-qubit gates.

4. The required qubits are those encoding the system, i.e., those that H acts upon, plus auxiliary qubits whose state $|G\rangle_a$ identifies the block in which H is encoded. The number of auxiliary qubits required depends on the choice of block encoding, but can be chosen to be $\lceil \log_2 T \rceil$ when the Hamiltonian contains T terms, as discussed in Section 2.1.

3.3   Other types of block encoding

The algorithm above can be applied for any block encoding of a linear combination of Pauli operators as in (3). The method is independent of how the indices of the Pauli operators are encoded in the auxiliary register, but the specifics of the measurement scheme do depend on the terms being Pauli operators. To see this, recall that when k is odd, we want to estimate the expectation value of the block encoding unitary U in the prepared state $|\psi_{\lfloor k/2 \rfloor}\rangle$. When the terms are Pauli operators, we accomplish this by separately estimating each term using the lower circuit in Fig. 1.

When a different block encoding is the input, however, we cannot necessarily measure U by repeatedly measuring in Pauli bases. In such a case, since by assumption U is a unitary that we have a quantum circuit for, we can estimate its expectation value using a Hadamard test or equivalent (see for example [25]). This would replace the Pauli basis measurement in the odd k circuit in Fig. 1. Note that this is also an option in the case of a Pauli Hamiltonian, where it represents trading off increased circuit depth (for the Hadamard test or equivalent) for a lower number of repetitions (not having to repeat for all Pauli bases).

4   Error analysis

In this section we analyse the error scaling in our algorithm subject to finite sample noise and regularization of the overlap matrix involved in the generalized eigenvalue problem (10). Finite sample noise enters because the matrix elements of $\mathbf{H}$ and $\mathbf{S}$ are obtained from expectation values estimated by repeated measurements. This introduces a statistical error to the resulting energy estimates.

In addition, when executed on a real quantum computer any quantum algorithm will be subject to device noise. Since our error analysis applies to arbitrary perturbations of the measured quantities, it can also be used to obtain energy error bounds under such device noise, but since that is device specific, we focus on finite sample noise for simplicity.

The overlap matrix $\mathbf{S}$ must be regularized because the stability of the generalized eigenvalue problem (10) depends on $\mathbf{S}$ being well-conditioned. In practice the condition number of $\mathbf{S}$ grows exponentially with the Krylov space dimension D, which has been proven for a Krylov space generated by powers of the Hamiltonian [53], and appears to hold numerically for Chebyshev polynomials of the Hamiltonian. This means that the generalized eigenvalue problem must be regularized, which may be done by thresholding the eigenvalues of $\mathbf{S}$, i.e., projecting both $\mathbf{H}$ and $\mathbf{S}$ onto the span of the eigenvectors of $\mathbf{S}$ whose eigenvalues are above some threshold $\epsilon > 0$ [23, 37].

For Krylov spaces generated by real time evolutions, the resulting error in the ground state energy estimate was analyzed in [37] (related techniques for handling noise and regularization are discussed in [23]). We adapt their proof to our Krylov space generated by Chebyshev polynomials of the Hamiltonian, with the following result:

Theorem 1. *Let H be an $N \times N$ Hamiltonian with energies $E_0 \leq E_1 \leq \cdots \leq E_{N-1}$ in the range $[-1, 1]$. Suppose we compute the D-dimensional Krylov space spanned by (13) using the method in Section 3, yielding noisy estimates*

$$(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{S}}) = (\boldsymbol{H} + \boldsymbol{\Delta}_H, \boldsymbol{S} + \boldsymbol{\Delta}_S) \qquad (28)$$

of $(\boldsymbol{H}, \boldsymbol{S})$, for some Hermitian perturbations

$(\boldsymbol{\Delta}_H, \boldsymbol{\Delta}_S)$ *whose spectral norms are* (η_H, η_S)*. Denote the total noise rate as*

$$\eta := \sqrt{\eta_H^2 + \eta_S^2}. \tag{29}$$

Let

$$\gamma_0 = \langle E_0 | \psi_0 \rangle \tag{30}$$

be the overlap of the initial reference state $|\psi_0\rangle$ *with the true ground state* $|E_0\rangle$*. Let* $\widetilde{\boldsymbol{S}}$ *be regularized by projecting it (and* $\widetilde{\boldsymbol{H}}$*) onto the subspace spanned by its eigenvectors with eigenvalues above some threshold* $\epsilon > 0$*. Let* ϵ_{total} *be the sum of the eigenvalues of* $\boldsymbol{S}$ *discarded by regularizing it according to* ϵ *in the same way. Then provided the noise rate* η *is sufficiently small (small enough that the assumptions of Lemma B.1 can be satisfied), there exists a choice of threshold*

$$\epsilon = \Omega\left((D^4 \eta)^{\frac{1}{1+\alpha}}\right) \tag{31}$$

for some constant $0 \leq \alpha \leq 1/2$ *such that for any* $\delta > 0$*, the error* $\mathcal{E}$ *in the ground state energy estimate coming from the regularized version of the noisy problem* $(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{S}})$ *is bounded by*

$$\mathcal{E} \leq O\Bigg((D^4 \eta)^{\frac{1}{1+\alpha}} + \frac{\sqrt{\delta}\,\epsilon_{total}}{|\gamma_0|^2}$$
$$+ \delta + \frac{1}{|\gamma_0|^2}\left(1 + \frac{\delta}{2}\right)^{-D}\Bigg). \tag{32}$$

The proof is given in Appendix B, which also gives a more precise characterization of what "provided the noise rate η is sufficiently small" means. The noise rate is not required to be exponentially small in any of the problem parameters. Although (32) is given as an asymptotic scaling for simplicity, the two results it is based on (Lemma B.1 and Theorem 2 in Appendix B) give explicit bounds on the errors due to noise and thresholding, respectively, so the total error bound (32) could also be reformulated as an explicit (i.e., nonasymptotic) bound.

The result (32) may be interpreted as follows. The last term is the error due to the exact (i.e., noiseless and nonthresholded) Krylov space, which vanishes exponentially quickly with the Krylov space dimension D. The third term, δ, is an energy error tolerance that determines the rate of convergence of the last term. The error bound (32) holds

for any $\delta > 0$, i.e., δ is only relevant to the analysis, and does not actually impact the algorithm. It characterizes the rate of exponential decay of amplitudes of energies more than δ above the ground state energy, via the final term in (32). Hence if δ is chosen to be equal to the spectral gap Δ, then the third term in (32) can be removed, because in that case there are no excited states within δ of the ground state energy. However, if δ is chosen to be larger than the spectral gap, then although the ground state energy is approximated, the corresponding state will not approximate the true ground state in general: it will only approximate some arbitrary state in the low-energy subspace within δ of the ground state energy.

The second term in (32) is due to regularization of $\widetilde{\boldsymbol{S}}$ by ϵ, i.e., to the discarding of eigenspaces of $\widetilde{\boldsymbol{S}}$ with eigenvalues smaller than ϵ. Finally, the first term in (32) is due to finite sample noise. The factor of $D^{\frac{4}{1+\alpha}}$ in this term results from the proof technique of Theorem 2.7 in [37] (reproduced in our notation as Lemma B.1 in Appendix B), in which the authors note that this factor might be reduced or eliminated by an improved proof; we discuss this further below. Numerics also indicate linear scaling $\epsilon = \Theta(\eta)$ (i.e., $\alpha = 0$) if various heuristics are adopted to select an optimal threshold ϵ [note that this scaling is compatible with (31)]. Such heuristics involve starting at a high threshold and decreasing it as long as the energy estimate continues to converge.

To understand the implications of Theorem 1 in practice, we can begin by observing that since the eigenvalues of $\widetilde{\boldsymbol{S}}$ decrease exponentially, as mentioned above,

$$\epsilon_{\text{total}} = \Theta(\epsilon). \tag{33}$$

Also, when the noise η comes from finite numbers of samples to obtain each matrix element in $(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{S}})$,

$$\eta = \Theta(1/\sqrt{M}), \tag{34}$$

when M measurements are performed per matrix element.

Note that a complete expression for η would also include some dependence on D, since η is obtained from spectral norms of the perturbation matrices $(\boldsymbol{\Delta}_H$ and $\boldsymbol{\Delta}_S)$. However, η enters (32) in the first two terms via (31) and (33), and in the context of

this total energy error bound, no strong dependence on D is generally found in practice, whether coming from η or from the explicit factors in (31). While in the presence of noise and thresholding, it sometimes happens that the error increases over short ranges of D, these increases are relatively small and the overall behavior is still bounded below a monotonically decreasing envelope. See our numerics in Section 5 and the numerics in [23, 37], for example. Providing a rigorous proof of this property is an interesting direction for future work.

Finally, although [37] provided numerics suggesting that $\alpha = 1/4$ is a good value, that was for the original bound through which α entered their proof, and our numerics suggest that in practice, at least in the context of Theorem 1 for our Chebyshev polynomial subspace, $\alpha = 0$ is a better value [see Section 5]. Hence, using (31) with $\alpha = 0$, (33), and (34), we obtain

$$\epsilon_{\text{total}} = \Theta(\epsilon) = \Theta(\eta) = \Theta\left(\frac{1}{\sqrt{M}}\right), \qquad (35)$$

which summarizes the "in practice" relations between these quantities. Subject to the above assumptions, (32) becomes the "in practice" error bound

$$\mathcal{E} \leq O\left(\frac{1}{\sqrt{M}} + \frac{\sqrt{\delta}}{|\gamma_0|^2\sqrt{M}} + \delta + \frac{1}{|\gamma_0|^2}\left(1+\frac{\delta}{2}\right)^{-D}\right). \tag{36}$$

From (36), we can find the Krylov space dimension D required to reach error $\mathcal{E}$ in the ground state energy estimate. From the final term in (36), in the limit of small initial state overlap $|\gamma_0|$ and δ we obtain

$$D = \Theta\left(\frac{\log\frac{1}{|\gamma_0|} + \log\frac{1}{\mathcal{E}}}{\delta}\right). \qquad (37)$$

To simplify this scaling, note that since δ appears as a term in the error (36) whenever it is larger than the spectral gap Δ (as discussed above), if the target error $\mathcal{E}$ is larger than Δ, we must choose $\delta = O(\mathcal{E})$. Given this, there is no reason to choose it to be smaller, so we can choose $\delta = \Theta(\mathcal{E})$. Substituting this into (37) yields

$$D = \Theta\left(\frac{\log\frac{1}{|\gamma_0|} + \log\frac{1}{\mathcal{E}}}{\mathcal{E}}\right). \qquad (38)$$

If instead the target error $\mathcal{E}$ is smaller than Δ, then we should instead choose $\delta = \Delta$, since in that case the δ term in (36) (and (32)) vanishes as noted above. Hence in general

$$\delta = \Theta(\max(\mathcal{E}, \Delta)), \qquad (39)$$

so a general bound on the required Krylov space dimension D is

$$D = \Theta\left[\left(\log\frac{1}{|\gamma_0|} + \log\frac{1}{\mathcal{E}}\right)\min\left(\frac{1}{\mathcal{E}}, \frac{1}{\Delta}\right)\right]. \quad (40)$$

Recall that D is also the maximum circuit depth in terms of queries to the block encoding operators (see (26) and Fig. 1).

The first two terms in (36) can then provide an asymptotic lower bound on the number of measurements M. From the first term, we find that

$$M = \Omega\left(\frac{1}{\mathcal{E}^2}\right), \qquad (41)$$

which should be unsurprising since this is always the scaling obtained from algorithms based on repeated sampling. From the second term, we obtain a lower bound in terms of the initial state overlap $|\gamma_0|^2$:

$$M = \Omega\left(\frac{\delta}{\mathcal{E}^2|\gamma_0|^4}\right) = \Omega\left(\frac{1}{\mathcal{E}|\gamma_0|^4}\right), \qquad (42)$$

where the second equality follows using $\delta = \Theta(\mathcal{E})$ as above. Since M must satisfy both of these asymptotic lower bounds, we can combine them as

$$M = \Theta\left(\frac{1}{\mathcal{E}^2} + \frac{1}{\mathcal{E}|\gamma_0|^4}\right), \qquad (43)$$

the total number of measurements required per matrix element. Finally, since all matrix elements of $\widetilde{\mathbf{S}}$ and $\widetilde{\mathbf{H}}$ are calculated from the $2D$ expectation values (20), the total number of measurements required for the entire algorithm is

$$DM = \widetilde{\Theta}\left(\left(\frac{1}{\mathcal{E}^2} + \frac{1}{\mathcal{E}|\gamma_0|^4}\right)\min\left(\frac{1}{\mathcal{E}}, \frac{1}{\Delta}\right)\right), \quad (44)$$

where the $\widetilde{\Theta}$ indicates that the logarithmic factor in (40) has been suppressed.

For a nonasymptotic analysis of this scaling, one would need to account for covariance of the resulting matrix element estimates, but this is unnecessary to obtain the asymptotic scaling (44). The

first term in (36) might in principle also have some bearing on the number of measurements as a function of the Krylov space dimension, but as noted above, [37] points out that this scaling may be an artifact of their proof technique. Regardless, the number of measurements depends only polynomially on the Krylov space dimension as well as the problem parameters $|\gamma_0|$ and $\mathcal{E}$.

5 Numerical demonstrations

As a demonstration of our method, we classically simulated its Krylov spaces for some example Hamiltonians. For each Hamiltonian, we directly calculated the expectation values in (20), which are the quantities we would actually estimate on the quantum computer in a real experiment, then used them to construct the subspace matrices **H** and **S** as in (17) and (19). We simulated the effect of finite sample noise by adding Gaussian noise with various standard deviations to those expectation values. We then regularized the generalized eigenvalue problem by thresholding as described in the previous section.

For zero noise simulations, the threshold was set to 10^{-13}. For simulations with nonzero noise, the threshold was set proportional to the noise rate, with the constants being 30 for the spin models and 50 for molecules. These constants were chosen to eliminate most spurious eigenvalues due to ill-conditioning: such spurious eigenvalues appear as dramatic downward jumps in the energy estimates with increasing subspace dimension. In an actual experiment one would want to choose the thresholds by detecting such jumps automatically, but for the purpose of our simple demonstration it was enough to choose constants by hand that eliminated most of the jumps. Since some spurious eigenvalues remained, we further eliminated outliers by taking as our datapoints the means of the middle 10% of 100 independent runs at each noise rate and subspace dimension.

Results of these simulations are shown in Fig. 2 to 4. Fig. 2 shows energy error as a function of Krylov space dimension for the J1-J2 spin model on a 4×4 square lattice. The Hamiltonian of the
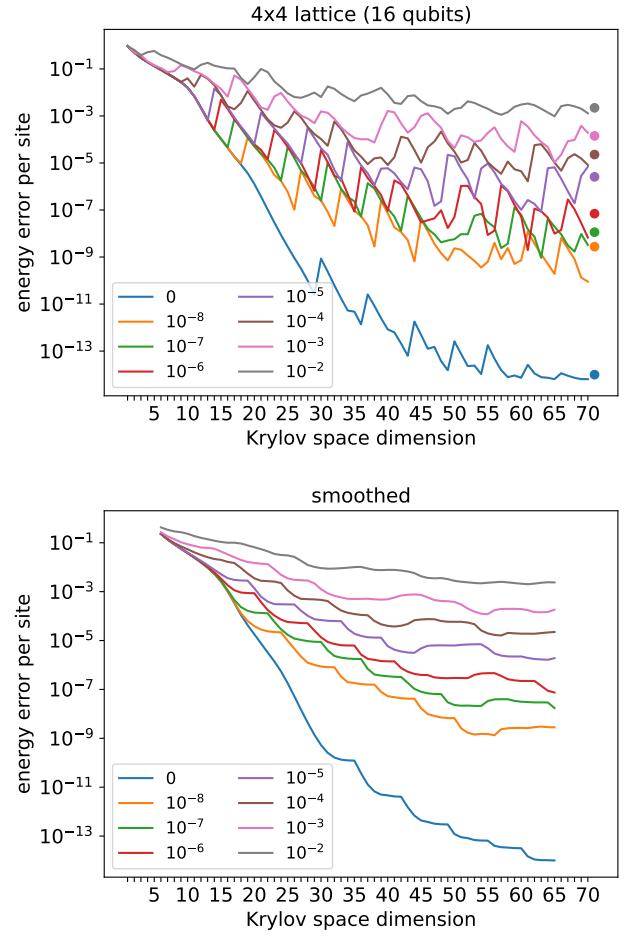

Figure 2: Results of numerical simulations of the nonorthogonalized Lanczos method implemented by our algorithm, for the J1-J2 model on a 4×4 square lattice with couplings $J_1 = 1$ and $J_2 = 0.5$. The upper panel shows energy error per site versus Krylov space dimension. The different curves correspond to different noise rates (given in the legend), which were imposed as Gaussian noise on the expectation values that would be estimated in the experiment [in (20)] to simulate the effect of finite sample noise. The points to the right of the curves are the means of the rightmost 10 points. Since the upper panel might be difficult to read, the lower panel represents the same data, but smoothed by taking the mean of each successive sequence of 10 points (each mean plotted above the center of its corresponding sequence).
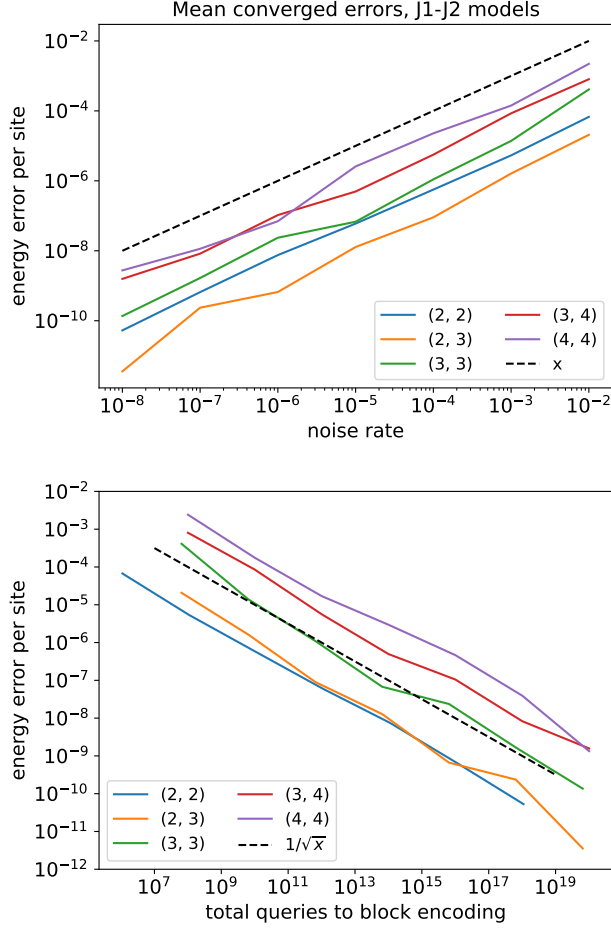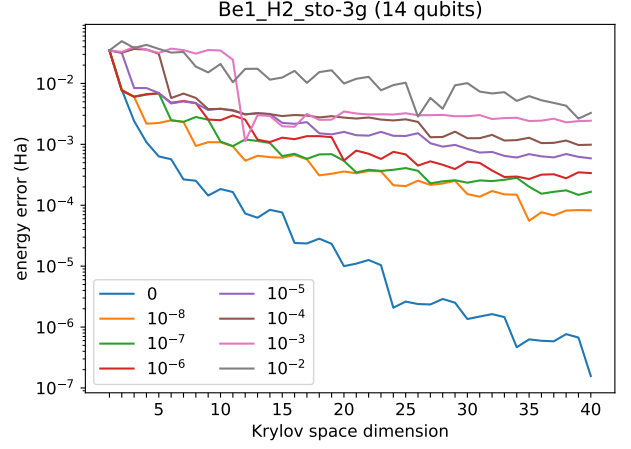
Figure 4: Results of numerical simulations of the nonorthogonalized Lanczos method implemented by our algorithm, for BeH_2 in the STO-3G basis, equilibrium configuration (bondlength 1.3264 Å). The plot shows energy error per site versus Krylov space dimension. As in Fig. 2, the different curves correspond to different noise rates (given in the legend), which were imposed as Gaussian noise on the expectation values that would be estimated in the experiment [in (20)] to simulate the effect of finite sample noise. This is an all-electron calculation (no frozen core).

Figure 3: Energy error per site versus noise rate per measured quantity (upper panel) and total queries to block encoding (lower panel) for the J1-J2 model on several different square lattices given in the legend. These energy errors were each averaged over a sequence of 10 dimensions once convergence was reached (i.e., once the contribution of the last term in (32) had become negligible); for example, the purple curve in the upper panel is the points on the right of the upper panel in Fig. 2, plotted against the corresponding noise rates. The dashed lines illustrate the expected scalings from the theoretical analysis. Note that the two panels show the same information since the total queries to the block encoding is proportional to the total number of shots times the circuit depth, which were chosen to be the depths required to reach convergence. For both plots above we took those depths to be 5 for $(2,2)$, 40 for $(2,3)$ and $(3,3)$, and 50 for $(3,4)$ and $(4,4)$.

J1-J2 model is given by

$$H = J_1 \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \vec{S}_i \cdot \vec{S}_j, \qquad (45)$$

where $\langle \cdot, \cdot \rangle$ denotes nearest-neighbor pairs, $\langle\langle \cdot, \cdot \rangle\rangle$ denotes next-nearest-neighbor pairs, and the couplings in our case were set to $J_1 = 1$ and $J_2 = 0.5$. The initial state was chosen to be the antiferromagnetic state, i.e., a checkerboard of alternating spins across the lattice, which yielded an initial state overlap of $|\gamma_0| = 0.179$. The upper panel shows the complete data, while for easier reading the lower panel shows a smoothed version of the same curves obtained by averaging each set of energy errors at 10 successive subspace dimensions. In both cases, after a period of ill-defined behavior at low Krylov space dimension, the overall trends at all noise rates settle into roughly exponential decay towards some minimum "converged" error level, with local fluctuations along the way. This is expected based on the theoretical error bounds in the previous section: the error decays exponentially with Krylov space dimension towards a minimum value set by

the first three terms in (32).

Fig. 3 shows energy error versus simulated noise rate for the J1-J2 model on a number of different square lattices. These energy errors were each averaged over a sequence of 10 dimensions once convergence was reached (i.e., once the contribution of the last term in (32) had become negligible). As we would expect, at this point energy error and noise rate are related approximately linearly (the dashed line shows the curve $y = x$).

Finally, Fig. 4 shows energy error versus Krylov space dimension for the electronic structure Hamiltonian BeH_2 in the STO-3G basis. The initial state was chosen to be the Hartree-Fock state, which has an initial state overlap of 0.986. Unfortunately, numerical instability in our classical simulation prevented us from reaching as high Krylov space dimension for the electronic structure Hamiltonian as for the spin models, and it is not clear that the energies had finished converging to their minimum values. This is possibly due to the choice of the Hartree-Fock state as the initial state: although it is a good initial approximation to the ground state in this case, the Hartree-Fock state contains no electron correlation. This means that all electron correlation of the lowest-energy state in the Krylov space must be contributed by the corresponding polynomial of the Hamiltonian, which may live in a poorly conditioned part of the space.

Since we have access only to classically simulable small models, we cannot draw strong conclusions from these simulations regarding the asymptotic behavior of our method. However, at least for these test cases, the behavior is roughly what we would expect from the theory in the previous section.

6 Conclusion

In this work we have shown how to exactly reproduce the Krylov space of the classical Lanczos method on a quantum computer, up to sampling error, while avoiding the exponential classical cost of representing quantum states. The required Krylov space dimension is inverse polynomial in the desired ground state energy error, and the sampling overhead is inverse polynomial in both the ground

state energy error and the initial state overlap. To our knowledge, the present method is conceptually the simplest use of block encoding in quantum algorithms to date, since it only requires applying up to one local basis rotation per circuit in addition to the block encoding unitaries themselves (see Fig. 1).

We conclude by discussing how the present method compares with existing quantum algorithms for approximating ground states. As noted in the introduction, if we compare to previous quantum subspace diagonalization methods [16, 17, 18, 21, 24, 25, 23], our algorithm benefits from being exactly equivalent to the classical Lanczos method up to sampling noise, without requiring approximate real or imaginary time evolution. This offers two advantages. First, the error in approximating real or imaginary time evolution, even if it can be suppressed, can never be reduced to zero. Second, in quantum algorithms for simulating real time evolution, the circuit depths increase as the target error is reduced. There are many methods to approximate time evolutions, but we mention two to illustrate this point. First, if the time evolutions are approximated by Trotterization, then they require depth of order $\epsilon^{-1/k}$ where k is the Trotter order and ϵ is the error. Hence deep circuits will be required merely to reach high accuracy in approximating the Krylov states: if d digits of accuracy are desired, then the resulting circuit depths scale as $O(\exp(d/r))$.

On the other hand, if the time evolutions are approximated by qubitization [38], the resulting circuit depths scale much more favorably with error as $O(\log(1/\epsilon))$. However, this case is easily compared to our algorithm, because qubitization uses the same block encoding unitaries that our method is based upon. Thus, while qubitization would require circuits using up to $O(D + \log(1/\epsilon))$ applications of the block encoding unitaries, as well as other controlled phases, to construct a Krylov space of dimension D, our method requires circuits of depth at most D in the same block encoding unitaries, with the only additional gates being the local basis rotation mentioned above.

This means that in practice, the advantage of our method over prior quantum subspace expansions is that it avoids all question of the accuracy

of approximation of the Krylov states themselves, completely eliminating this source of error and circuit depth. This reduces both depth and number of measurements required to obtain a result of a given accuracy. Since circuit depth and number of measurements are key barriers to the success of quantum algorithms and will remain so even into the fault-tolerant era, this advance is significant.

In comparison to using quantum imaginary time evolution to approximate the ground state, we do not require the underlying quantum state to have finite correlation length as it goes through the evolution. Even in the presence of a finite correlation length, there are significant prefactors in imaginary time evolution that lead to high computational cost (true, the cost is $4^{O(\log n)} = O(\text{poly}(n))$, but this can be a high-degree polynomial [16]). Adiabatic state preparation [9] requires a gapped path from some efficiently-solvable Hamiltonian to the target Hamiltonian, while our method is independent of the spectral gap when that is small (see Section 4). It also avoids the typical pitfalls of parametric optimization that appear in variational quantum eigensolvers [2], such as local minima and barren plateaus [54, 55, 56, 57, 58], and admits rigorous error bounds (Section 4).

The most well-known algorithm for quantum computation of ground state energies is phase estimation. To be successful, both our method and phase estimation require an initial state overlap $|\gamma_0| = \Omega(1/\text{poly}(n))$, for different reasons. For phase estimation, the state overlap determines the success probability of the algorithm, while for our method it impacts the number of measurements required to obtain small errors (only entering the circuit depth logarithmically). Similar to other quantum subspace diagonalization methods, our algorithm has quantifiable robustness to noise in the sense that it produces energy estimates with bounded error even if the subspace matrices are not exact. This robustness is not encountered in conventional phase estimation (although more recent ground-state preparation algorithms such as [28] do possess some noise resilience). On the other hand, phase estimation achieves Heisenberg scaling with error, i.e., $O(1/\mathcal{E})$, while our algorithm possesses the typical $O(1/\mathcal{E}^2)$ scaling of methods with finite sample noise. Hence, although phase estimation

may be preferable to reach extremely high precision energy estimates in the long run, many specific problems of interest have fixed error demands depending on their application. For such problems, in the intermediate term while robustness to noise is still the crucial provision, our method presents a useful option.

Acknowledgements

References

[1] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. "Simulated quantum computation of molecular energies". Science **309**, 1704–1707 (2005).

[2] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". Nature Communications **5**, 4213 EP – (2014).

[3] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. "Scalable quantum simulation of molecular energies". Phys. Rev. X **6**, 031007 (2016).

Accepted in ⟨ ⟩uantum 2023-04-22, click title to verify. Published under CC-BY 4.0.

13

[4] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets". Nature **549**, 242–246 (2017).

[5] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum computational chemistry". Rev. Mod. Phys. **92**, 015003 (2020).

[6] Yuan Su, Dominic W. Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. "Fault-tolerant quantum simulations of chemistry in first quantization". PRX Quantum **2**, 040332 (2021).

[7] Alexei Y. Kitaev. "Quantum measurements and the abelian stabilizer problem" (1995). url: doi.org/10.48550/arXiv.quant-ph/9511026.

[8] Seth Lloyd. "Universal quantum simulators". Science **273**, 1073–1078 (1996).

[9] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. "Quantum computation by adiabatic evolution" (2000). url: doi.org/10.48550/arXiv.quant-ph/0001106.

[10] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. "Simulating physical phenomena by quantum networks". Phys. Rev. A **65**, 042323 (2002).

[11] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. "Simulation of electronic structure hamiltonians using quantum computers". Molecular Physics **109**, 735–750 (2011).

[12] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. "Low-depth quantum simulation of materials". Phys. Rev. X **8**, 011044 (2018).

[13] Jarrod R. McClean, Mollie E. Kimchi-Schwartz, Jonathan Carter, and Wibe A. de Jong. "Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states". Phys. Rev. A **95**, 042308 (2017).

[14] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. "Computation of molecular spectra on a quantum processor with an error-resilient algorithm". Phys. Rev. X **8**, 011021 (2018).

[15] William J Huggins, Joonho Lee, Unpil Baek, Bryan O'Gorman, and K Birgitta Whaley. "A non-orthogonal variational quantum eigensolver". New Journal of Physics **22**, 073009 (2020).

[16] Mario Motta, Chong Sun, Adrian T. K. Tan, Matthew J. O'Rourke, Erika Ye, Austin J. Minnich, Fernando G. S. L. Brandão, and Garnet Kin-Lic Chan. "Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution". Nature Physics **16**, 205–210 (2020).

[17] Robert M. Parrish and Peter L. McMahon. "Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation" (2019). url: doi.org/10.48550/arXiv.1909.08925.

[18] Nicholas H. Stair, Renke Huang, and Francesco A. Evangelista. "A multireference quantum krylov algorithm for strongly correlated electrons". Journal of Chemical Theory and Computation **16**, 2236–2245 (2020).

[19] Tyler Takeshita, Nicholas C. Rubin, Zhang Jiang, Eunseok Lee, Ryan Babbush, and Jarrod R. McClean. "Increasing the representation accuracy of quantum simulations of chemistry without extra quantum resources". Phys. Rev. X **10**, 011004 (2020).

[20] Lin Lin and Yu Tong. "Near-optimal ground state preparation". Quantum **4**, 372 (2020).

[21] Jeffrey Cohn, Mario Motta, and Robert M. Parrish. "Quantum filter diagonalization with compressed double-factorized hamiltonians". PRX Quantum **2**, 040352 (2021).

[22] Nobuyuki Yoshioka, Hideaki Hakoshima, Yuichiro Matsuzaki, Yuuki Tokunaga, Yasunari Suzuki, and Suguru Endo. "General-

ized quantum subspace expansion". Phys. Rev. Lett. **129**, 020502 (2022).

[23] Katherine Klymko, Carlos Mejuto-Zaera, Stephen J. Cotton, Filip Wudarski, Miroslav Urbanek, Diptarka Hait, Martin Head-Gordon, K. Birgitta Whaley, Jonathan Moussa, Nathan Wiebe, Wibe A. de Jong, and Norm M. Tubman. "Real-time evolution for ultracompact hamiltonian eigenstates on quantum hardware". PRX Quantum **3**, 020323 (2022).

[24] Kazuhiro Seki and Seiji Yunoki. "Quantum power method by a superposition of time-evolved states". PRX Quantum **2**, 010333 (2021).

[25] Cristian L. Cortes and Stephen K. Gray. "Quantum krylov subspace algorithms for ground- and excited-state energy estimation". Phys. Rev. A **105**, 022417 (2022).

[26] Unpil Baek, Diptarka Hait, James Shee, Oskar Leimkuhler, William J. Huggins, Torin F. Stetina, Martin Head-Gordon, and K. Birgitta Whaley. "Say no to optimization: A non-orthogonal quantum eigensolver" (2022). url: doi.org/10.48550/arXiv.2205.09039.

[27] Nikolay V. Tkachenko, Yu Zhang, Lukasz Cincio, Alexander I. Boldyrev, Sergei Tretiak, and Pavel A. Dub. "Quantum davidson algorithm for excited states" (2022). url: doi.org/10.48550/arXiv.2204.10741.

[28] Lin Lin and Yu Tong. "Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers". PRX Quantum **3**, 010318 (2022).

[29] Yulong Dong, Lin Lin, and Yu Tong. "Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices". PRX Quantum **3**, 040305 (2022).

[30] Alexei Kitaev, A. H. Shen, and M. N. Vyalyi. "Classical and quantum computation". Volume 47 of Graduate Studies in Mathematics. American Mathematical Society, Providence, RI. (2002).

[31] Julia Kempe, Alexei Kitaev, and Oded Regev. "The complexity of the local hamiltonian problem". SIAM Journal on Computing **35**, 1070–1097 (2006).

[32] Cornelius Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". Journal of Research of the National Bureau of Standards **45**, No. 4 (1950).

[33] Jane K Cullum and Ralph A Willoughby. "Lanczos algorithms for large symmetric eigenvalue computations: Vol. i: Theory". Society for Industrial and Applied Mathematics. (2002).

[34] Shmuel Kaniel. "Estimates for some computational techniques in linear algebra". Mathematics of Computation **20**, 369–378 (1966).

[35] Christopher Conway Paige. "The computation of eigenvalues and eigenvectors of very large sparse matrices". PhD thesis. University of London. (1971). url: ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.307848.

[36] Y. Saad. "On the rates of convergence of the lanczos and the block-lanczos methods". SIAM Journal on Numerical Analysis **17**, 687–706 (1980).

[37] Ethan N. Epperly, Lin Lin, and Yuji Nakatsukasa. "A theory of quantum subspace diagonalization". SIAM Journal on Matrix Analysis and Applications **43**, 1263–1290 (2022).

[38] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization". Quantum **3**, 163 (2019).

[39] Guang Hao Low and Isaac L. Chuang. "Optimal hamiltonian simulation by quantum signal processing". Phys. Rev. Lett. **118**, 010501 (2017).

[40] Iordanis Kerenidis and Anupam Prakash. "Quantum Recommendation Systems". In Christos H. Papadimitriou, editor, 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Volume 67 of Leibniz International Proceedings in Informatics (LIPIcs), pages 49:1–49:21. Dagstuhl, Germany (2017).

Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[41] Mark Steudtner and Stephanie Wehner. "Estimating exact energies in quantum simulation without toffoli gates". Phys. Rev. A **101**, 052329 (2020).

[42] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. "Explicit quantum circuits for block encodings of certain sparse matrices" (2022). url: doi.org/10.48550/arXiv.2203.10236.

[43] Daan Camps and Roel Van Beeumen. "Fable: Fast approximate quantum circuits for block-encodings". In 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). Pages 104–113. (2022).

[44] Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. "Quantum fingerprinting". Phys. Rev. Lett. **87**, 167902 (2001).

[45] Dorit Aharonov, Vaughan Jones, and Zeph Landau. "A polynomial quantum algorithm for approximating the jones polynomial". In Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing. Pages 427–436. New York, NY, USA (2006). Association for Computing Machinery.

[46] Charles Hadfield, Sergey Bravyi, Rudy Raymond, and Antonio Mezzacapo. "Measurements of quantum hamiltonians with locally-biased classical shadows". Communications in Mathematical Physics **391**, 951–967 (2022).

[47] Hsin-Yuan Huang, Richard Kueng, and John Preskill. "Efficient estimation of pauli observables by derandomization". Phys. Rev. Lett. **127**, 030503 (2021).

[48] Bujiao Wu, Jinzhao Sun, Qi Huang, and Xiao Yuan. "Overlapped grouping measurement: A unified framework for measuring quantum states". Quantum **7**, 896 (2023).

[49] Stefan Hillmich, Charles Hadfield, Rudy Raymond, Antonio Mezzacapo, and Robert Wille. "Decision diagrams for quantum measurements with shallow circuits". In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). Pages 24–34. (2021).

[50] Charles Hadfield. "Adaptive pauli shadows for energy estimation" (2021). url: doi.org/10.48550/arXiv.2105.12207.

[51] Ariel Shlosberg, Andrew J. Jena, Priyanka Mukhopadhyay, Jan F. Haase, Felix Leditzky, and Luca Dellantonio. "Adaptive estimation of quantum observables". Quantum **7**, 906 (2023).

[52] Tzu-Ching Yen, Aadithya Ganeshram, and Artur F. Izmaylov. "Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates". npj Quantum Information **9**, 14 (2023).

[53] Bernhard Beckermann and Alex Townsend. "On the singular values of matrices with displacement structure". SIAM Journal on Matrix Analysis and Applications **38**, 1227–1248 (2017).

[54] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nature Communications **9**, 4812 (2018).

[55] Alexey Uvarov and Jacob Biamonte. "On barren plateaus and cost function locality in variational quantum algorithms". Journal of Physics A: Mathematical and Theoretical (2021).

[56] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". Nature Communications **12**, 1791 (2021).

[57] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. "Noise-induced barren plateaus in variational quantum algorithms". Nature Communications **12**, 6961 (2021).

[58] Junyu Liu, Zexi Lin, and Liang Jiang. "Laziness, barren plateau, and

noise in machine learning" (2022). url: doi.org/10.48550/arXiv.2206.09313.

[59] Å. Björck. "Numerical methods in matrix computations". Texts in Applied Mathematics. Springer International Publishing. (2014).

[60] Daniel Gottesman. "Stabilizer codes and quantum error correction" (1997). url: doi.org/10.48550/arXiv.quant-ph/9705052.

[61] Emanuel Knill and Raymond Laflamme. "Theory of quantum error-correcting codes". Phys. Rev. A **55**, 900–911 (1997).

A Proof of Lemma 1

The following lemma is based directly on Section IV in [38]:

Lemma 1 (Chebyshev polynomials from block encoding). *Given a block encoding (U, G) of a Hamiltonian H, such that $U^2 = \mathbb{1}$, let*

$$R := (2|G\rangle_a\langle G|_a - \mathbb{1}_a) \otimes \mathbb{1}_s \qquad (46)$$

be the reflection around $|G\rangle_a$ in the auxiliary space. Then

$$(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes \mathbb{1}_s) = T_k(H) \qquad (47)$$

for any $k = 0, 1, 2, ...,$ where $T_k(\cdot)$ is the kth Chebyshev polynomial of the first kind. In other words, $(RU)^k$ is a block encoding of $T_k(H)$.

Proof. Let $|\lambda\rangle$ denote an arbitrary eigenvector of H with eigenvalue λ. We can evaluate the action of U on $|G\rangle_a \otimes |\lambda\rangle$ as follows:

$$U(|G\rangle_a \otimes |\lambda\rangle) = |G\rangle_a \otimes H|\lambda\rangle + \eta_\lambda |\perp_\lambda\rangle, \quad (48)$$

where η_λ is some positive number that preserves normalization and $|\perp_\lambda\rangle$ is some (normalized) state of all of the qubits that is orthogonal to $|G\rangle_a$ in the auxiliary space, i.e., such that

$$(\langle G|_a \otimes \mathbb{1}_s)|\perp_\lambda\rangle = 0. \qquad (49)$$

Hence, (48) simply represents resolving the state $U(|G\rangle_a \otimes |\lambda\rangle)$ into its components parallel and orthogonal to $|G\rangle_a$ in the auxiliary space: the latter

$(|\perp_\lambda\rangle)$ is unknown, but the former is

$$\begin{aligned}(|G\rangle_a\langle G|_a &\otimes \mathbb{1}_s)U(|G\rangle_a \otimes |\lambda\rangle) \\ &= (|G\rangle_a \otimes \mathbb{1}_s)(\langle G|_a \otimes \mathbb{1}_s)U(|G\rangle_a \otimes |\lambda\rangle) \qquad (50) \\ &= (|G\rangle_a \otimes \mathbb{1}_s)(\mathbb{1}_a \otimes H|\lambda\rangle) = |G\rangle_a \otimes H|\lambda\rangle\end{aligned}$$

by (1), which forms the first term in (48).

Since $|\lambda\rangle$ is an eigenstate of H, (48) becomes

$$\begin{aligned}U(|G\rangle_a \otimes |\lambda\rangle) &= |G\rangle_a \otimes \lambda|\lambda\rangle + \eta_\lambda |\perp_\lambda\rangle \\ &= \lambda|G\rangle_a \otimes |\lambda\rangle + \sqrt{1 - \lambda^2}|\perp_\lambda\rangle,\end{aligned}$$
$$(51)$$

where the second step follows by identifying $\eta_\lambda = \sqrt{1 - \lambda^2}$ as the required normalization factor for the second term. Hence, if we formally denote

$$|G\rangle_a \otimes |\lambda\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |\perp_\lambda\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad (52)$$

we may write

$$U\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda \\ \sqrt{1 - \lambda^2} \end{pmatrix}. \qquad (53)$$

Note that the vectors in (52) represent the states of the "virtual qubit" that is referred to in the name *qubitization*.

We also know that $U^2 = \mathbb{1}$ (5), which implies that within the two-dimensional subspace spanned by (52), the matrix representing U^2 must be

$$U^2 \to \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \qquad (54)$$

From this and (53), it follows that the matrix form of U in this two-dimensional subspace must be

$$U \to \begin{pmatrix} \lambda & \sqrt{1 - \lambda^2} \\ \sqrt{1 - \lambda^2} & -\lambda \end{pmatrix}, \qquad (55)$$

since (53) defines the first column of this matrix, and the second column is required to make it self-inverse.

The matrix for U in (55) is a reflection. We can transform it into a rotation if we can flip the signs of the entries in the second row, since the result will have the form

$$U_y(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \qquad (56)$$

Accepted in 〈 𝔮uantum 2023-04-22, click title to verify. Published under CC-BY 4.0.

17

for a rotation angle θ defined by

$$\cos\theta = \lambda, \quad \sin\theta = \sqrt{1-\lambda^2}. \tag{57}$$

In order to flip the signs in the second row of (55), we need to multiply on the left by Z within the two-dimensional subspace:

$$
\begin{aligned}
Z &\begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix} \\
&= \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ -\sqrt{1-\lambda^2} & \lambda \end{pmatrix},
\end{aligned} \tag{58}
$$

which has the desired form (56).

One way to implement the action of Z on the two-dimensional subspace spanned by (52) is to reflect around $|G\rangle_a$ in the auxiliary space, i.e., to implement R as defined in (46). The actions of R on the subspace are

$$
\begin{aligned}
R \begin{pmatrix} 1 \\ 0 \end{pmatrix} &= (2|G\rangle_a\langle G|_a - \mathbb{1}_a)(|G\rangle_a \otimes |\lambda\rangle) \\
&= |G\rangle_a \otimes |\lambda\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
R \begin{pmatrix} 0 \\ 1 \end{pmatrix} &= (2|G\rangle_a\langle G|_a - \mathbb{1}_a)|\perp_\lambda\rangle \\
&= -|\perp_\lambda\rangle = -\begin{pmatrix} 0 \\ 1 \end{pmatrix},
\end{aligned} \tag{59}
$$

where the second equation follows from (49). Hence, the matrix representation of R within the two-dimensional subspace is

$$R \to \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z. \tag{60}$$

Therefore, within the two-dimensional subspace

$$RU \to \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ -\sqrt{1-\lambda^2} & \lambda \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{61}$$

where θ is defined by (57).

As an aside, note that the form (46) for R is not actually a necessary condition, although it is sufficient. All that we strictly require is that R

implements a reflection around $|G\rangle_a$ within each two-dimensional subspace associated to each λ: its action on the remainder of the auxiliary space is irrelevant. We will exploit this fact later on in constructing efficient implementations of R.

We can now use the fact that k powers of a rotation by angle θ are simply the rotation by angle $k\theta$, so

$$(RU)^k \to \begin{pmatrix} \cos k\theta & \sin k\theta \\ -\sin k\theta & \cos k\theta \end{pmatrix}. \tag{62}$$

But

$$\cos\theta = \lambda \quad \Rightarrow \quad \cos k\theta = T_k(\lambda), \tag{63}$$

where $T_k(\cdot)$ is the kth Chebyshev polynomial of the first kind, so

$$(RU)^k \to \begin{pmatrix} T_k(\lambda) & \sqrt{1-T_k^2(\lambda)} \\ -\sqrt{1-T_k^2(\lambda)} & T_k(\lambda) \end{pmatrix}. \tag{64}$$

By the definition (52) of the two-dimensional subspace associated to λ, this implies that

$$
\begin{aligned}
(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes |\lambda\rangle) &= T_k(\lambda)|\lambda\rangle \\
&= T_k(H)|\lambda\rangle.
\end{aligned} \tag{65}
$$

The final step in the proof is to note that any arbitrary state $|\psi\rangle$ of the system can be expressed as a superposition of Hamiltonian eigenstates:

$$|\psi\rangle = \sum_\lambda \beta_\lambda|\lambda\rangle, \tag{66}$$

so

$$
\begin{aligned}
&(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes \mathbb{1}_s)\,|\psi\rangle \\
&= \sum_\lambda \beta_\lambda(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes |\lambda\rangle) \\
&= \sum_\lambda \beta_\lambda T_k(H)|\lambda\rangle = T_k(H)|\psi\rangle.
\end{aligned} \tag{67}
$$

Since this applies for any state $|\psi\rangle$, we have

$$(\langle G|_a \otimes \mathbb{1}_s)(RU)^k(|G\rangle_a \otimes \mathbb{1}_s) = T_k(H), \tag{68}$$

as desired.

$\square$

B Convergence of the noisy, thresholded Krylov method with Chebyshev polynomials

In this section we prove Theorem 1, which is stated in the main text and bounds the error of our method subject to noise and regularization via eigenvalue thresholding.

Theorem 1. *Let H be an $N \times N$ Hamiltonian with energies $E_0 \leq E_1 \leq \cdots \leq E_{N-1}$ in the range $[-1, 1]$. Suppose we compute the D-dimensional Krylov space spanned by (13) using the method in Section 3, yielding noisy estimates*

$$(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{S}}) = (\boldsymbol{H} + \boldsymbol{\Delta}_H, \boldsymbol{S} + \boldsymbol{\Delta}_S) \qquad (69)$$

of $(\boldsymbol{H}, \boldsymbol{S})$, for some Hermitian perturbations $(\boldsymbol{\Delta}_H, \boldsymbol{\Delta}_S)$ whose spectral norms are (η_H, η_S). Denote the total noise rate as

$$\eta := \sqrt{\eta_H^2 + \eta_S^2}. \qquad (70)$$

Let

$$\gamma_0 = \langle E_0 | \psi_0 \rangle \qquad (71)$$

be the overlap of the initial reference state $|\psi_0\rangle$ with the true ground state $|E_0\rangle$. Let $\widetilde{\boldsymbol{S}}$ be regularized by projecting it (and $\widetilde{\boldsymbol{H}}$) onto the subspace spanned by its eigenvectors with eigenvalues above some threshold $\epsilon > 0$. Let ϵ_{total} be the sum of the eigenvalues of $\boldsymbol{S}$ discarded by regularizing it according to ϵ in the same way. Then provided the noise rate η is sufficiently small (small enough that the assumptions of Lemma B.1 can be satisfied), there exists a choice of threshold

$$\epsilon = \Omega\left((D^4\eta)^{\frac{1}{1+\alpha}}\right) \qquad (72)$$

for some constant $0 \leq \alpha \leq 1/2$ such that for any $\delta > 0$, the error $\mathcal{E}$ in the ground state energy estimate coming from the regularized version of the noisy problem $(\widetilde{\boldsymbol{H}}, \widetilde{\boldsymbol{S}})$ is bounded by

$$\mathcal{E} \leq O\bigg((D^4\eta)^{\frac{1}{1+\alpha}} + \frac{\sqrt{\delta}\,\epsilon_{total}}{|\gamma_0|^2}$$

$$+ \delta + \frac{1}{|\gamma_0|^2}\left(1 + \frac{\delta}{2}\right)^{-D}\bigg). \qquad (73)$$

Proof. Let E_0 be the true ground state energy of H, let E_0' be the energy estimate from the noiseless, thresholded Krylov space, and let $\widetilde{E}_0'$ be the energy estimate from the noisy, thresholded Krylov space. Then the total error is $\mathcal{E} = |\widetilde{E}_0' - E_0|$, for which we obtain the bound (73) via the triangle inequality as a first step:

$$\mathcal{E} = \left|\widetilde{E}_0' - E_0\right| \leq \left|\widetilde{E}_0' - E_0'\right| + \left|E_0' - E_0\right|. \qquad (74)$$

We will upper bound the first term, characterizing the error due to noise, using Theorem 2.7 of [37], which can be applied directly to our case. We will upper bound the second term in (74) using Theorem 2, proved below, which is based on Theorem 3.1 in [37], but which we had to modify to apply to our case, since the version in [37] is specific to subspaces generated by real time evolutions.

Theorem 2.7 in [37] may be restated in our notation as follows:

Lemma B.1 (Theorem 2.7 in [37]). *Instate the prevailing notation. Let $(\lambda_i, \boldsymbol{v}_i)$ be the eigenpairs of $\boldsymbol{S}$, ordered by eigenvalue from largest to smallest. Let m be the greatest index such that $\lambda_m > \epsilon$. Four assumptions are required for this theorem to hold:*

1. *(gap at thresholding level) Assume that the eigenvalues of $\boldsymbol{S}$ are sufficiently separated at the threshold level and the noise rate is low enough that*

$$\lambda_{m+1} + \eta_S \leq \epsilon. \qquad (75)$$

This assumption is merely to ensure that during thresholding the same numbers of eigenvalues are discarded from the noisy and noiseless overlap matrices, so that the resulting thresholded matrix problems have the same dimensions.

2. *(bound on $\boldsymbol{H}$ relative to $\boldsymbol{S}$) Assume that*

$$\left|\boldsymbol{v}_i^\dagger \boldsymbol{H} \boldsymbol{v}_j\right| \leq \mu \min(\lambda_i, \lambda_j)^{1-\alpha} \max(\lambda_i, \lambda_j)^\alpha \qquad (76)$$

Accepted in 〈 〉uantum 2023-04-22, click title to verify. Published under CC-BY 4.0.

19

for all i, j, for some constants $\mu > 0$ and $0 \leq \alpha \leq 1/2$. This bound always holds if $\alpha = 1/2$, but smaller α may be more realistic in practice (see [37]).

3. *(bound on noise rate relative to threshold) Let $\rho > 0$ be such that $(1+\rho)\epsilon \leq \lambda_m$. Assume that*

$$D^4\chi \leq \epsilon, \tag{77}$$

where χ is defined as

$$\chi := 3(2+\mu)\left(1+\frac{1}{\rho}\right)\left(\frac{\|\mathbf{S}\|}{\epsilon}\right)^\alpha \eta_S + \eta_H. \tag{78}$$

4. *(gap condition) Assume that*

$$\arctan(E_1') - \arctan(E_0') \geq \arcsin\left(\frac{D^4\chi}{\epsilon}\right), \tag{79}$$

where E_0', E_1' are the lowest and second-to-lowest eigenvalues of the noiseless, thresholded problem.

Then the difference between the lowest eigenvalue E_0' of the noiseless thresholded problem and the lowest eigenvalue $\widetilde{E}_0'$ of the noisy thresholded problem is bounded as

$$\left|\arctan(E_0') - \arctan(\widetilde{E}_0')\right| \leq \arcsin(\kappa D^4\chi), \tag{80}$$

where κ is the condition number of $\arctan(E_0')$. [Note: Theorem 2.7 in [37] also includes the assumption

$$\left(1+\frac{1}{\rho}\right)\eta_S \leq \epsilon, \tag{81}$$

but this is guaranteed to hold by (77).]

Our input Hamiltonian will be normalized such that its eigenvalues lie inside the range $[-1, 1]$. Therefore, the condition number κ of $\arctan(E_0')$ is bounded as

$$\min_{x\in[-1,1]}\left|\frac{x\arctan'(x)}{\arctan(x)}\right| \leq \kappa \leq \max_{x\in[-1,1]}\left|\frac{x\arctan'(x)}{\arctan(x)}\right|$$

$$\Rightarrow \quad \min_{x\in[-1,1]}\left|\frac{x}{(1+x^2)\arctan(x)}\right| \leq \kappa \leq \max_{x\in[-1,1]}\left|\frac{x}{(1+x^2)\arctan(x)}\right| \tag{82}$$

$$\Rightarrow \quad \left|\frac{1}{2\arctan(1)}\right| \leq \kappa \leq \lim_{x\to 0}\frac{x}{(1+x^2)\arctan(x)} = \lim_{x\to 0}\frac{x}{\arctan(x)}$$

$$\Rightarrow \quad \frac{2}{\pi} \leq \kappa \leq 1.$$

Hence we can replace κ by 1 in (80), and the resulting bound will be weaker but equivalent in scaling:

$$\left|\arctan(E_0') - \arctan(\widetilde{E}_0')\right| \leq \arcsin(D^4\chi), \tag{83}$$

Also, for $x \in [-1, 1]$, the minimum value of the first derivative of $\arctan(x)$ is $1/2$, so

$$\frac{1}{2}|E_0' - \widetilde{E}_0'| \leq \left|\arctan(E_0') - \arctan(\widetilde{E}_0')\right|. \tag{84}$$

Finally, for $x \in [0, 1]$, $\arcsin(x) \leq \pi x/2$, so we can obtain a simplified energy bound of

$$|E_0' - \widetilde{E}_0'| \leq \pi D^4\chi. \tag{85}$$

Similarly, since the maximum value of the first derivative of $\arctan(x)$ on $[-1, 1]$ is 1 and $\arcsin(x) \geq x$ on $[0, 1]$, the condition (79) on the gap can be simplified to

$$\Delta' := E_1' - E_0' \geq \frac{D^4\chi}{\epsilon}. \tag{86}$$

Hence we just need to choose

$$\epsilon \geq \frac{D^4\chi}{\Delta'} \tag{87}$$

and (77) will also be guaranteed, so the only remaining assumption in the statement of Lemma B.1 will be that (75) holds.

Inserting the definition (78) of χ into (87) yields

$$\epsilon = \Theta\left(\left(\frac{D^4\eta}{\Delta'}\right)^{\frac{1}{1+\alpha}}\right). \tag{88}$$

If we now insert the definition of χ into (85) and simplify within asymptotic notation, we can obtain

$$|E_0' - \tilde{E}_0'| \leq O\left(\frac{D^4\eta}{\epsilon^\alpha}\right) = O\left((D^4\eta)^{\frac{1}{1+\alpha}}(\Delta')^{\frac{\alpha}{1+\alpha}}\right), \tag{89}$$

where the second step uses (88). Since Δ' will be smaller than 1 except possibly at the lowest Krylov space dimensions, we can obtain a weaker upper bound by simply dropping the factor that depends on Δ', obtaining

$$|E_0' - \tilde{E}_0'| \leq O\left((D^4\eta)^{\frac{1}{1+\alpha}}\right); \tag{90}$$

in practice, the inclusion of Δ' in the denominator of the lower bound to ϵ as in (88) is probably unrealistic anyway, so we are likely not sacrificing any tightness of the bound by making this simplification. Inserting (90) in (74) provides the first term in (73), while (88) yields the condition (72).

The remaining terms in (73), coming from the second term in (74), are provided by (91), which is the result of Theorem 2, below. When inserting this into (74), we assume that the threshold ϵ is small enough that the denominator in (91) scales as $\Theta(|\gamma_0|^2)$, allowing us to ignore the second term the denominator in the asymptotic scaling. This is why the statement of Theorem 1 only claims that if the noise rate is sufficiently small then a threshold exists such that (73) is satisfied. □

Theorem 2. *Instate the prevailing notation. The error in the noiseless thresholded ground state en-*

ergy estimate E_0' is bounded as

$$0 \leq E_0' - E_0$$

$$\leq \delta + 8\frac{\sqrt{\delta}\,\epsilon_{total} + (1 - \gamma^2 + 4\epsilon_{total})\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2}, \tag{91}$$

where

$$\gamma := \sqrt{\sum_{i\ s.t.\ E_i - E_0\ \leq\ \delta} |\gamma_i|^2} \tag{92}$$

is the overlap of the initial state with the subspace whose energy is within δ of E_0. [Note: this form of the bound only holds when $|\gamma_0| > \sqrt{2D}\epsilon$, and is only tight when the denominator in (91) is $\Omega(|\gamma_0|^2)$.]

Proof. Let $\mathbf{K}$ be the *Krylov matrix* whose columns are the Krylov vectors, i.e.,

$$\mathbf{K} = \begin{bmatrix} |\psi_0\rangle & T_1(H)|\psi_0\rangle & T_2(H)|\psi_0\rangle & \cdots & T_k(H)|\psi_0\rangle \end{bmatrix}, \tag{93}$$

where

$$k := D - 1. \tag{94}$$

The projected Hamiltonian $\mathbf{H}$ and overlap matrix $\mathbf{S}$ are given by

$$\mathbf{H} = \mathbf{K}^\dagger H \mathbf{K}, \quad \mathbf{S} = \mathbf{K}^\dagger \mathbf{K}. \tag{95}$$

Hence, the eigenvalues of $\mathbf{S}$ are the squares of the singular values of $\mathbf{K}$, and the corresponding eigenvectors of $\mathbf{S}$ are the right singular vectors of $\mathbf{K}$. Therefore, applying thresholding that removes the eigenspaces of $\mathbf{S}$ with eigenvalues smaller than ϵ is equivalent to truncating the Krylov space to the span of the left singular vectors (which are states in the Hilbert space) of $\mathbf{K}$ with singular values at least $\sqrt{\epsilon}$.

We can factor $\mathbf{K}$ as follows:

$$\mathbf{K} = \underbrace{\begin{bmatrix} |E_0\rangle & |E_1\rangle & |E_2\rangle & \cdots & |E_{N-1}\rangle \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} \gamma_0 & & & & \\ & \gamma_1 & & & \\ & & \gamma_2 & & \\ & & & \ddots & \\ & & & & \gamma_{N-1} \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} 1 & T_1(E_0) & \cdots & T_k(E_0) \\ 1 & T_1(E_1) & \cdots & T_k(E_1) \\ 1 & T_1(E_2) & \cdots & T_k(E_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & T_1(E_{N-1}) & \cdots & T_k(E_{N-1}) \end{bmatrix}}_{F}. \tag{96}$$

Let $\mathbf{K}' = \mathbf{\Psi}\mathbf{\Gamma}\mathbf{F}'$ denote the matrix obtained by setting all singular values of $\mathbf{K}$ smaller than $\sqrt{\epsilon}$ to zero.

Let R' be the range (column space) of $\mathbf{K}'$, i.e., the thresholded Krylov space, spanned by the left singular vectors of $\mathbf{K}$ with singular values at least $\sqrt{\epsilon}$. Then by the Rayleigh-Ritz variational principle, the error in the ground state energy estimate in the thresholded Krylov space

$$E_0' - E_0 = \min_{|\psi\rangle \in R' \setminus \{\mathbf{0}\}} \frac{\langle \psi | (H - E_0) | \psi \rangle}{\langle \psi | \psi \rangle}. \tag{97}$$

We will upper bound the above minimization by inserting a particular state $|\psi\rangle \in R'$ into its argument.

To construct this state, we use the polynomial p^* given in Lemma B.2, below, with the parameters a, b, d in Lemma B.2 set to $a = \delta^2/4$, $b = 1$, and $d = \lfloor k/2 \rfloor$: in terms of this, define the function f as

$$f(x) := p^*\left(\frac{(x - E_0)^2}{4}\right). \tag{98}$$

By Lemma B.2,

$$f(E_0) = p^*(0) = 1, \tag{99}$$

but for any $x \in [-1, 1]$ such that $|x - E_0| \geq \delta$,

$$|f(x)| \leq 2\left(1 + \sqrt{\frac{a}{b}}\right)^{-d} = 2\left(1 + \frac{\delta}{2}\right)^{-\lfloor k/2 \rfloor}. \tag{100}$$

Note that this bound is only tight when it is much less than 1. Since p^* has degree $\lfloor k/2 \rfloor$, $f(x)$ has degree either $k - 1$ or k. Let c_j be the coefficients of the Chebyshev expansion of f:

$$f(x) = \sum_{j=0}^{k} c_j T_j(x). \tag{101}$$

Since $|p^*(x)| \leq 1$ for $0 \leq x \leq 1$, $|f(x)| \leq 1$ for $-1 \leq x \leq 1$, and for x outside the interval $[E_0 - \delta, E_0 + \delta]$, the much tighter bound (100)

holds. Therefore,

$$\int_{-1}^{1} dx \frac{f^2(x)}{\sqrt{1 - x^2}}$$
$$\leq \int_{E_0 - \delta}^{E_0 + \delta} dx \frac{1}{\sqrt{1 - x^2}}$$
$$\quad + \int_{[-1,1] \setminus [E_0 - \delta, E_0 + \delta]} dx \frac{4\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}}{\sqrt{1 - x^2}}$$
$$\leq \int_{1 - 2\delta}^{1} dx \frac{1}{\sqrt{1 - x^2}} + \int_{-1}^{1 - 2\delta} dx \frac{4\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}}{\sqrt{1 - x^2}}$$
$$\leq \cos^{-1}(1 - 2\delta)$$
$$\quad + 4\left(\pi - \cos^{-1}(1 - 2\delta)\right)\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}$$
$$\leq \pi\sqrt{\delta} + 4(\pi - 2\sqrt{\delta})\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}, \tag{102}$$

which holds for $\delta < 1/4$, so since

$$\int_{-1}^{1} dx \frac{T_i(x) T_j(x)}{\sqrt{1 - x^2}} = \begin{cases} \pi & \text{if } i = j = 0, \\ \pi/2 & \text{if } i = j \neq 0, \\ 0 & \text{if } i \neq j, \end{cases} \tag{103}$$

we obtain the following bound on the coefficients c_j:

$$\sum_{j=0}^{k} c_j^2 \leq c_0^2 + \sum_{j=0}^{k} c_j^2 = \frac{2}{\pi} \int_{-1}^{1} dx \frac{f^2(x)}{\sqrt{1 - x^2}}$$
$$\leq 2\sqrt{\delta} + 8\left(1 - \frac{2}{\pi}\sqrt{\delta}\right)\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}$$
$$:= g(\delta, k). \tag{104}$$

The above is really just an application of Parseval's Theorem, using the fact that the Chebyshev polynomials are orthogonal on $[-1, 1]$ with respect to the measure $dx/\sqrt{1 - x^2}$. As noted after (100), this upper bound $g(\delta, k)$ is only tight when its second term is much smaller than 1, which will always hold in our regime of interest, since as we will see, this second term is proportional to the main energy error term we will end up with.

Let $|\psi\rangle$ be defined in terms of the coefficients c_j as

$$|\psi\rangle = \sum_{j=0}^{k} c_j \sum_{i=0}^{N-1} \gamma_i \mathbf{F}_{ij}' |E_i\rangle \in R' \tag{105}$$

(note that this state is in R' because $\sum_{i=0}^{N-1} \gamma_i \mathbf{F}'_{ij}|E_i\rangle$ is the jth column in $\mathbf{K}'$.) Inserting this into (97) yields

$$E'_0 - E_0 \leq \frac{\sum_{i=1}^{N-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{\sum_{i=0}^{N-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2}.$$

$$(106)$$

Let I be the least value of i such that $E_I - E_0 > \delta$. We separately upper bound the parts of (106) coming from terms in the numerator with $i \geq I$ and $i < I$. We start with the former, i.e., let $i \geq I$ and consider the term

$$\frac{|\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{\sum_{l=0}^{N-1} |\gamma_l|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{lj}\right|^2}$$

$$\leq \frac{|\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{|\gamma_0|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{0j}\right|^2}.$$

$$(107)$$

We upper bound the numerator as follows, using (100) and (104):

$$\left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 = \left|\sum_{j=0}^{k} c_j (\mathbf{F}'_{ij} - T_j(E_i)) + \sum_{j=0}^{k} c_j T_j(E_i)\right|^2$$

$$= \left|\sum_{j=0}^{k} c_j (\mathbf{F}'_{ij} - T_j(E_i)) + f(E_i)\right|^2$$

$$\leq 2 \left(\sum_{j=0}^{k} |c_j|^2\right) \left(\sum_{j=0}^{k} |\mathbf{F}'_{ij} - T_j(E_i)|^2\right) + 2|f(E_i)|^2$$

$$\leq 2g(\delta, k) \left(\sum_{j=0}^{k} \alpha_{ij}^2\right) + 4 \left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor},$$

$$(108)$$

where in the last line we defined

$$\alpha_{ij} := |\mathbf{F}'_{ij} - T_j(E_i)|. \qquad (109)$$

For the denominator of this term, we use

$$\left|\sum_{j=0}^{k} c_j \mathbf{F}'_{0j}\right| = \left|\sum_{j=0}^{k} c_j T_j(E_0) + \sum_{j=0}^{k} c_j (\mathbf{F}'_{0j} - T_j(E_0))\right|$$

$$\geq |f(E_0)| - \left|\sum_{j=0}^{k} c_j (\mathbf{F}'_{0j} - T_j(E_0))\right|$$

$$\geq 1 - \sum_{j=0}^{k} |c_j| \left|\mathbf{F}'_{0j} - T_j(E_0)\right|$$

$$\geq 1 - 2 \sum_{j=0}^{k} \alpha_{0j},$$

$$(110)$$

where the second line follows from the reverse triangle inequality and (101), the third line follows from the triangle inequality and (99), and the last line follows from the fact that each $|c_j| \leq 2$ (obtained by evaluating the integral in the first line of (104) replacing $f^2(x)$ with its upper bound of 1). Extending the above bound,

$$\left|\sum_{j=0}^{k} c_j \mathbf{F}'_{0j}\right| \geq 1 - 2 \sqrt{(k+1) \sum_{j=0}^{k} \alpha_{0j}^2}$$

$$\geq 1 - \frac{2}{|\gamma_0|} \sqrt{(k+1)\epsilon},$$

$$(111)$$

using the spectral norm bound

$$\sqrt{\epsilon} \geq \|\mathbf{K}' - \mathbf{K}\| = \|\mathbf{\Gamma}(\mathbf{F}' - \mathbf{F})\|$$

$$\geq \sqrt{|\gamma_0|^2 \sum_{j=0}^{k} |\mathbf{F}'_{0j} - T_j(E_0)|^2} = |\gamma_0| \sqrt{\sum_{j=0}^{k} \alpha_{0j}^2}.$$

$$(112)$$

Hence, the term (107) is upper bounded by

$$\frac{2|\gamma_i|^2 (E_i - E_0) \left(g(\delta,k) \sum_{j=0}^{k} \alpha_{ij}^2 + 2 \left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}\right)}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2}$$

$$\leq \frac{4|\gamma_i|^2 \left(g(\delta,k) \sum_{j=0}^{k} \alpha_{ij}^2 + 2 \left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}\right)}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2}.$$

$$(113)$$

Now consider together all of the terms in (106) with $i < I$:

$$\frac{\sum_{i=0}^{I-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{\sum_{l=0}^{N-1} |\gamma_l|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{lj}\right|^2}$$

$$\leq \frac{\sum_{i=0}^{I-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{\sum_{i=0}^{I-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2}. \tag{114}$$

The right-hand side of (114) is an expectation value of the difference between energy and ground state energy for some state in the subspace whose energies are within δ of the ground state energy. Hence by the Rayleigh-Ritz variational principle, (114) is upper bounded by δ:

$$\frac{\sum_{i=0}^{I-1} |\gamma_i|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{ij}\right|^2 (E_i - E_0)}{\sum_{l=0}^{N-1} |\gamma_l|^2 \left|\sum_{j=0}^{k} c_j \mathbf{F}'_{lj}\right|^2} \leq \delta. \tag{115}$$

Inserting (113) and (115) into (106), we obtain

$$E'_0 - E_0$$

$$\leq \delta + \sum_{i=I}^{N-1} \frac{4|\gamma_i|^2 \left(g(\delta,k) \sum_{j=0}^{k} \alpha_{ij}^2 + 2\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}\right)}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2} \, is$$

$$= \delta + 4 \frac{g(\delta,k)\epsilon_{\text{total}} + 2(1-\gamma^2)\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2}$$

$$\leq \delta + 8 \frac{\sqrt{\delta}\,\epsilon_{\text{total}} + (1-\gamma^2 + 4\epsilon_{\text{total}})\left(1 + \frac{\delta}{2}\right)^{-2\lfloor k/2 \rfloor}}{\left(|\gamma_0| - 2\sqrt{(k+1)\epsilon}\right)^2}, \tag{116}$$

our desired result, where the last line is obtained by inserting the definition of $g(\delta,k)$ as in (104) and upper bounding $1 - \frac{2}{\pi}\sqrt{\delta} \leq 1$, and in the second line we inserted

$$\gamma := \sqrt{\sum_{i=0}^{I-1} |\gamma_i|^2} = \sqrt{\sum_{i \text{ s.t. } E_i - E_0 \leq \delta} |\gamma_i|^2} \tag{117}$$

and ϵ_{total}, which we define to be the squared Frobenius distance from $\mathbf{K}$ to $\mathbf{K}'$:

$$\epsilon_{\text{total}} := \|\mathbf{K}' - \mathbf{K}\|_F^2 = \|\mathbf{\Gamma}(\mathbf{F}' - \mathbf{F})\|_F^2 = \sum_{i=0}^{N-1} \sum_{j=0}^{k} |\gamma_i|^2 \alpha_{ij}^2 \tag{118}$$

(the second equality follows because $\mathbf{\Psi}$ is unitary.) Since $\mathbf{K}' - \mathbf{K}$ is a matrix whose only nonzero singular values are those singular values of $\mathbf{K}$ that are smaller than $\sqrt{\epsilon}$ and are thus dropped in thresholding, and the Frobenius norm may also be written

$$\epsilon_{\text{total}} = \|\mathbf{K}' - \mathbf{K}\|_F^2 = \text{Tr}\left((\mathbf{K}' - \mathbf{K})^\dagger (\mathbf{K}' - \mathbf{K})\right), \tag{119}$$

ϵ_{total} as defined by (118) is equal to the sum of squares of the singular values of $\mathbf{K}$ that are discarded in thresholding. Since $\mathbf{S} = \mathbf{K}^\dagger \mathbf{K}$, ϵ_{total} is the sum of eigenvalues of $\mathbf{S}$ that are discarded in thresholding, as claimed in the theorem statement. $\square$

Lemma B.2 ([59],Theorem 4.1.11). *Let $0 < a < b$ and let Π_d^* be the space of residual polynomials (polynomials whose value at 0 is 1) of degree at most d. The solution to*

$$\beta(a,b,d) = \min_{p \in \Pi_d^*} \max_{x \in [a,b]} |p(x)| \tag{120}$$

is

$$p^*(x) = \frac{T_d\left(\frac{b+a-2x}{b-a}\right)}{T_d\left(\frac{b+a}{b-a}\right)}, \tag{121}$$

and the corresponding minimal value is

$$\beta(a,b,d) = T_d^{-1}\left(\frac{b+a}{b-a}\right)$$

$$\leq 2\left(1 + \sqrt{\frac{a}{b}}\right)^{-d}. \tag{122}$$

Note that the upper bound is only tight when it is much less than 1.

Proof. Eq. (121) is the result given in Theorem 4.1.11 in [59]. To obtain (122), note that for $x \in [a,b]$,

$$-1 \leq \frac{b+a-2x}{b-a} \leq 1, \tag{123}$$

so

$$\left|T_d\left(\frac{b+a-2x}{b-a}\right)\right| \leq 1 \tag{124}$$

with equality at either end of the domain (123). Hence

$$\beta(a,b,d) = \min_{p\in\Pi_k^*}\max_{x\in[a,b]}|p(x)|$$

$$= \max_{x\in[a,b]}\left|\frac{T_d\left(\frac{b+a-2x}{b-a}\right)}{T_d\left(\frac{b+a}{b-a}\right)}\right| = T_d^{-1}\left(\frac{b+a}{b-a}\right)$$

$$\leq 2\left(\frac{b+a}{b-a}+\sqrt{\left(\frac{b+a}{b-a}\right)^2-1}\right)^{-d}$$

$$= 2\left(\frac{b-a}{b+a}\right)^d\left(1+\sqrt{1-\left(\frac{b-a}{b+a}\right)^2}\right)^{-d} \quad (125)$$

$$\leq 2\left(1+\sqrt{1-\left(\frac{b-a}{b+a}\right)^2}\right)^{-d}$$

$$\leq 2\left(1+\sqrt{\frac{a}{b}}\right)^{-d},$$

where the last step follows after some straightforward but tedious algebra. □

C Block encoding based on binary representation of Pauli operators

C.1 Block encoding components

Recall that to implement a block encoding, we need a preparation procedure for the state

$$|G\rangle_a = \sum_{i=0}^{T-1}\sqrt{\alpha_i}|i\rangle_a \quad (126)$$

and an implementation of the unitary

$$U = \sum_{i=0}^{T-1}|i\rangle_a\langle i|_a\otimes P_i. \quad (127)$$

Suppose we let

$$|i\rangle_a = |\vec{x},\vec{z}\rangle_a, \quad (128)$$

where $(\vec{x},\vec{z})$ is the binary representation of P_i [60, 61]. In this representation, both $\vec{x}$ and $\vec{z}$ are binary vectors, where a 1 in the jth position in $\vec{x}$ indicates that the Pauli operator contains an X on qubit j, and a 1 in the jth position in $\vec{z}$ indicates that the Pauli operator contains an Z on qubit j (so

presence of both indices Y). In other words, the Pauli operator that corresponds to $(\vec{x},\vec{z})$ is

$$P(\vec{x},\vec{z}) = \bigotimes_{j=0}^{n-1} i^{x_jz_j}X^{x_j}Z^{z_j} \quad (129)$$

(the i in the above expression is the imaginary number, not the index of the Pauli operator).

Hence, this block encoding requires $2n$ auxiliary qubits for n system qubits, so it triples the qubit requirement compared to the system alone. In exchange, the gate costs are substantially lower, depending on the specific Hamiltonian to be encoded, so this block encoding option may be advantageous on near-term quantum computers whose noisy gates make shorter depth and fewer controls a higher priority than fewer qubits. We can now express U as

$$U = \sum_{\vec{x},\vec{z}}|\vec{x},\vec{z}\rangle_a\langle\vec{x},\vec{z}|_a\otimes P(\vec{x},\vec{z}). \quad (130)$$

Inserting (129), we can factor this as

$$U = \prod_{j=0}^{n-1} i^{x_jz_j}\left(\text{ctrl}_{x_j}\text{-}X_j\right)\left(\text{ctrl}_{z_j}\text{-}Z_j\right), \quad (131)$$

so U can be implemented as only three layers of singly-controlled operations: a first layer that applies Z_j to each qubit controlled on the corresponding auxiliary qubit encoding z_j, a second layer that applies X_j to each qubit controlled on the corresponding auxiliary qubit encoding x_j, and a third layer that implements the two-qubit controlled phase

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix} \quad (132)$$

on each pair of auxiliary qubits x_j, z_j. A circuit diagram for U is shown in Fig. 5.

The complexity of preparing

$$|G\rangle_a = \sum_{i=0}^{T-1}\sqrt{\alpha_i}|(\vec{x},\vec{z})_i\rangle_a, \quad (133)$$

where $(\vec{x},\vec{z})_i$ is the binary representation of the ith term in the Hamiltonian, depends strongly on

form of the Hamiltonian. For each term P_i in the Hamiltonian, the locality l of P_i (i.e., the number of qubits it acts upon) determines the Hamming weight of $(\vec{x}, \vec{z})_i$, since each of $\vec{x}$ and $\vec{z}$ can have Hamming weight at most l. Hence, one case in which this encoding may be advantageous is when the input Pauli Hamiltonian is local, i.e., l is constant.

For example, the $l = 2$ case includes nearest-neighbor spin models. In this case, the Hamiltonian is composed of terms of the form

$$\alpha_1 \sigma_j \quad \text{or} \quad \alpha_2 \sigma_j \sigma_k, \tag{134}$$

where α_1, α_2 are coefficients and

$$\sigma_j(x_j, z_j) = i^{x_j z_j} X_j^{x_j} Z_j^{z_j} \tag{135}$$

denotes a single-qubit Pauli operator acting on qubit j. The corresponding terms in $|G\rangle_a$ will be

$$\begin{aligned}\sqrt{\alpha_1}|00...x_j...0, 00...z_j...0\rangle, \\ \sqrt{\alpha_2}|00...x_j...x_k...0, 00...z_j...z_k...0\rangle,\end{aligned} \tag{136}$$

in which the only potentially nonzero bits x_j, x_k, z_j, z_k are in positions j and k in their respective bitstrings.

In order to prepare such a state, it is enough if we can prepare an arbitrary real superposition of computational basis states with Hamming weight one or two. For weight one, we can accomplish this as follows:

$$\begin{aligned} |000...00\rangle &\xrightarrow{X_0} |100...00\rangle \\ &\xrightarrow{\text{PSWAP}_{01}(\theta_1)} |010...00\rangle \\ &\xrightarrow{\text{PSWAP}_{12}(\theta_2)} |001...00\rangle \\ &\quad\vdots \\ &\xrightarrow{\text{PSWAP}_{n-2,n-1}(\theta_{n-1})} |000...01\rangle, \end{aligned} \tag{137}$$

where after each step we only show the newest state in the superposition, and $\text{PSWAP}_{ij}(\theta)$ is defined as the two-qubit gate

$$\text{PSWAP}_{ij}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{138}$$

acting on qubits i and j. If we desired to obtain the coefficient $\beta_i \geq 0$ for the bitstring with the one in the ith position, then the angles in (137) should be chosen as follows:

$$\cos\theta_1 = \beta_1 \;\Rightarrow\; \theta_1 = \arccos\beta_1,$$

$$\sqrt{1 - \beta_1^2}\cos\theta_2 = \beta_2 \;\Rightarrow\; \theta_2 = \arccos\left(\frac{\beta_2}{\sqrt{1 - \beta_1^2}}\right),$$

$$\vdots$$

$$\sqrt{1 - \sum_{j=1}^{i-1}\beta_j^2}\cos\theta_i = \beta_i$$

$$\Rightarrow\; \theta_i = \arccos\left(\frac{\beta_i}{\sqrt{1 - \sum_{j=1}^{i-1}\beta_j^2}}\right). \tag{139}$$

This procedure requires only linear qubit connectivity.

We can prepare an arbitrary real superposition of weight-two bitstrings as follows:

$$\begin{aligned} |000...00\rangle &\xrightarrow{X_0 X_1} |1100000...0\rangle \\ &\xrightarrow{\text{C}_0\text{-PSWAP}_{12}(\theta_1)} |1010000...0\rangle \\ &\xrightarrow{\text{C}_2\text{-PSWAP}_{01}(\theta_2)} |0110000...0\rangle \\ &\xrightarrow{\text{C}_2\text{-PSWAP}_{13}(\theta_3)} |0011000...0\rangle \\ &\xrightarrow{\text{C}_3\text{-PSWAP}_{21}(\theta_4)} |0101000...0\rangle \\ &\xrightarrow{\text{C}_3\text{-PSWAP}_{10}(\theta_5)} |1001000...0\rangle \\ &\xrightarrow{\text{C}_0\text{-PSWAP}_{34}(\theta_6)} |1000100...0\rangle \\ &\xrightarrow{\text{C}_4\text{-PSWAP}_{01}(\theta_7)} |0100100...0\rangle \\ &\xrightarrow{\text{C}_4\text{-PSWAP}_{12}(\theta_8)} |0010100...0\rangle \\ &\xrightarrow{\text{C}_4\text{-PSWAP}_{23}(\theta_9)} |0001100...0\rangle \\ &\xrightarrow{\text{C}_4\text{-PSWAP}_{35}(\theta_10)} |0000110...0\rangle \\ &\xrightarrow{\text{C}_5\text{-PSWAP}_{43}(\theta_11)} |0001010...0\rangle \\ &\xrightarrow{\text{C}_5\text{-PSWAP}_{32}(\theta_12)} |0010010...0\rangle \\ &\xrightarrow{\text{C}_5\text{-PSWAP}_{21}(\theta_13)} |0100010...0\rangle \\ &\quad\vdots \end{aligned} \tag{140}$$

where again the state shown after each step is

only the latest state to be added to the superposition and one can check that each controlled partial swap acts nontrivially on only the state obtained in the preceding step. The controlled partial swap C_k-$\text{PSWAP}_{ij}(\theta)$ is simply the partial swap defined in (138), controlled on qubit k. Hence, if β_i is the desired coefficient of the ith bitstring obtained in the order in (140), the associated angle θ_i is again simply given by (139). The above procedure requires arbitrary qubit connectivity, but this can be changed to linear connectivity by adding $O(n)$ SWAPs per step in (140) for n qubits.

In general, to prepare states containing terms of the forms (136), we need to be able to prepare an arbitrary real superposition of terms of Hamming weight one and two. Let $\beta_i^{(1)}$ be the desired coefficients for the weight-one terms and $\beta_i^{(2)}$ be the desired coefficients for the weight-two terms. To prepare the desired state, first prepare the weight-one part, only with $\beta_0^{(1)}$ replaced by

$$\sqrt{(\beta_0^{(1)})^2 + \sum_i (\beta_i^{(2)})^2} \qquad (141)$$

so that in the final state the term $|100...0\rangle$ has the above coefficient. Then apply the partial CNOT

$$\text{CNOT}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & \sin\theta & -\cos\theta \end{pmatrix} \qquad (142)$$

to qubits 0 and 1 with the angle

$$\theta = \arccos\left(\frac{\beta_0^{(1)}}{\sqrt{(\beta_0^{(1)})^2 + \sum_i (\beta_i^{(2)})^2}} \right). \qquad (143)$$

This will map the $|100...0\rangle$ term to

$$\beta_0^{(1)}|100...0\rangle + \sqrt{\sum_i (\beta_i^{(2)})^2}|110...0\rangle. \qquad (144)$$

Finally, apply the procedure (140) to prepare the weight-two part of the superposition, skipping the initial $X_0 X_1$ step. Since all of the C-PSWAPs do not change states of weight one, this will not disturb the already prepared weight-one part of the superposition, but will simply distribute the amplitude $\sqrt{\sum_i (\beta_i^{(2)})^2}$ placed on $|110...0\rangle$ over the other weight-two states, as desired.

The above procedures suffice to prepare any superpositions of weight-one and weight-two bitstrings, which capture all terms of the form (136) corresponding to X, Z, XX, ZZ, or XZ terms in the Hamiltonian. To include terms containing Ys, we need to obtain ones in the same locations in the $\vec{x}$ and $\vec{z}$ registers. This can be accomplished by doubly- and triply-controlled partial CNOTs, starting from states prepared as above and using a similar strategy to distribute amplitude.

The one remaining piece we require for the block encoding is to implement the signs of the terms. Recall that the unitary U in (127) is supposed to implement the signed Pauli operators so that the coefficients α_i in the Hamiltonian can be nonnegative. However, our implementation of U above simply applied each Pauli without a sign. To include the signs, we can first assume without loss of generality that all single-qubit terms in the Hamiltonian are Zs with positive signs (this amounts to a choice of $|0\rangle$ and $|1\rangle$ states for each qubit). Hence, we only have to implement signs on the interaction terms.

If the only interaction terms have the form XX, ZZ, or XZ, then the sign for each term can be implemented as a controlled-Z on the pair of auxiliary qubits that are one in the binary representation of the term. This requires at most T controlled-Z gates for T terms. If terms containing Ys are also present, then these phases must be triply-controlled, i.e., four-qubit gates instead. A circuit diagram for G is given in Fig. 6.

C.2 Reflection about block encoding state

The above completes the construction of the block encoding itself, but implementing qubitization as in Lemma 1 also requires reflections around $|G\rangle_a$. Our construction so far requires at most four-qubit gates, so we would prefer to avoid the strategy of simply inverting G, reflecting around $|0\rangle_a$, and then reapplying G, since this requires a phase controlled on the entire auxiliary register. However, in the case of spin models, we can use the fact (pointed out in the proof of Lemma 1) that we do not in fact require a complete reflection around $|G\rangle_a$, but only within the subspace containing states of the auxiliary qubits that can be reached by applying U

to $|G\rangle_a$.

U does not change the states of the auxiliary qubits directly, since it is only controlled by them, although this does mean that it entangles them with the system qubits. In particular, U does not change the Hamming weight of the auxiliary states. To reflect around $|G\rangle_a$, we must invert G, i.e., apply $G^\dagger$ to the auxiliary qubits. This operation can increase the Hamming weight of each of the registers $|\vec{x}\rangle$ and $|\vec{z}\rangle$ to at most three, which we can see as follows. Prior to applying $G^\dagger$, the Hamming weights of $|\vec{x}\rangle$ and $|\vec{z}\rangle$ are each at most two. The only operation that can increase the Hamming weight higher than this is the partial CNOT in (142) applied to qubits 0 and 1 in each register to divide amplitude between the weight-one and weight-two terms. When the inverse of this operation is applied to a state of weight two, if qubit 0 is in state $|1\rangle$ and qubit 1 is in state $|0\rangle$, while the other $|1\rangle$ is somewhere else, then one of the terms resulting from the partial CNOT will have weight three, with qubit 1 now contributing the third $|1\rangle$. However, the only other operation to be applied to this state that does not conserve Hamming weight is the initial X_0 (137), which decreases the weight of the state. Of course, this X_0 may increase the weight of other computational basis states in the superposition, but none of those could have weight higher than two prior to its application, so overall it is impossible to end up with states of weight higher than three.

Given this, after inverting G, we only need to reflect around $|0\rangle_a$, the all-zeroes state of the auxiliary qubits, in the subspace spanned by states of Hamming weight up to three (for both registers $|\vec{x}\rangle$ and $|\vec{z}\rangle$). This allows us to avoid reflecting by applying a phase controlled on the entire register. Instead, we can simply use two additional auxiliary qubits to count the ones in $|\vec{x}\rangle$ and another two additional auxiliary qubits to count the ones in $|\vec{z}\rangle$. For example, to count the ones in $|\vec{x}\rangle$, we prepare its two auxiliary qubits in the state

$$|w_1^{(x)}, w_0^{(x)}\rangle = |00\rangle, \tag{145}$$

where $w_0^{(x)}$ and $w_1^{(x)}$ refer to the zeroth and first bits of the weight of $|\vec{x}\rangle$, respectively. Then for each qubit in $|\vec{x}\rangle$, we apply Toffoli on $w_1^{(x)}$ controlled

on $w_0^{(x)}$ and the current qubit in $|\vec{x}\rangle$, followed by CNOT on $w_0^{(x)}$ controlled on the current qubit in $|\vec{x}\rangle$. Since there are at most three ones in $|\vec{x}\rangle$, this stores the number of ones in $|\vec{x}\rangle$ in $|w_1^{(x)}, w_0^{(x)}\rangle$ as a binary number. Therefore, once we have done this for both $|\vec{x}\rangle$ and $|\vec{z}\rangle$, we can simply apply a -1 phase controlled on the resulting four auxiliary qubits being in state $|0000\rangle$. After that, we uncompute the auxiliary qubits used for counting, then reapply G, and our reflection about $|G\rangle_a$ is complete. A circuit diagram for R is given in Fig. 9.

C.3 Analysis

Finally, we can analyze the costs of U, G, and R. For simplicity, we will do this for the case where the Hamiltonian contains only Z, ZZ, and XX terms, i.e., it is a Heisenberg XYZ model with the YY interaction set to zero. For U, we require two main steps:

1. We argued above that the cost prior to implementing the signs of the terms is three layers of singly-controlled operations as in (131), for a total of $3n$ CNOTs and CPHASEs.

2. To implement the signs of the terms, we require at most T CZs for T terms (one CZ per term with a negative sign).

Hence the total cost to implement U is

$$3n + T \tag{146}$$

two-qubit gates.

For G, we also require two main steps, as outlined above:

1. For each of $|\vec{x}\rangle$ and $|\vec{z}\rangle$, prepare the weight-one part of the superposition as in (137). This requires a single NOT followed by $n-1$ PSWAPs for each, so in total requires 2 NOTs and $2(n-1)$ PSWAPs.

2. For each of $|\vec{x}\rangle$ and $|\vec{z}\rangle$, prepare the weight-one part of the superposition as in (140), replacing the first step (the double NOT) with the partial CNOT as in (142). This requires the

partial CNOT followed by

$$\sum_{i=2}^{n-2} i = \frac{1}{2}n^2 - \frac{3}{2}n \qquad (147)$$

C-PSWAPs, so in total requires 2 PCNOTs and $n^2 - 3n$ C-PSWAPs.

We obtain the total cost for G by adding up the above gate counts, but first we note that the three qubit gates, which are all C-PSWAPS, can be implemented in place via four two-qubit gates each (this construction is inspired by the contruction of controlled time-evolutions for certain Hamiltonians in [29]). For

$$\hat{\mu}_{jk} := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad (148)$$

where j, k denote the qubits acted upon, we can express a partial swap as

$$\mathrm{PSWAP}_{jk}(\theta) = e^{i\theta\hat{\mu}_{jk}}. \qquad (149)$$

We also have

$$\hat{\mu}_{jk} Z_j = -Z_j \hat{\mu}_{jk}, \qquad (150)$$

so

$$\mathrm{C}_j\text{-PSWAP}_{kl}(\theta) = e^{i\theta\hat{\mu}_{kl}/2} \cdot C_j Z_k \cdot e^{-i\theta\hat{\mu}_{kl}/2} \cdot C_j Z_k, \qquad (151)$$

a product of four two-qubit gates, as desired. Using this, the total cost for G comes to

$$\begin{aligned} 4n^2 - 10n & \quad \text{two-qubit,} \\ 2 & \quad \text{single-qubit} \end{aligned} \qquad (152)$$

gates (for $n \geq 3$).

Finally, implementing R requires an application of $G^\dagger$ and an application of G, with the reflection about $|0\rangle_a$ in between. The cost of the latter is $4n$ Toffoli gates and $4n$ CNOTs (to compute and uncompute the weight of the state), plus the four-qubit CPHASE. As in the implementation of G, we can replace each three qubit gate (Toffolis in this case) with four two-qubit gates, this time using

$$\hat{\nu}_{jk} := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad (153)$$

to generate a CNOT as

$$\mathrm{CNOT}_{jk} = e^{i\pi\hat{\nu}_{jk}/2}S_j^\dagger, \qquad (154)$$

where

$$S_j := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad (155)$$

acts on the controlling qubit. As above, we also have

$$\hat{\nu}_{jk} Z_j = -Z_j \hat{\nu}_{jk}, \qquad (156)$$

Therefore,

$$\mathrm{Toffoli}_{jkl} = e^{i\pi\hat{\nu}_{kl}/4} \cdot C_j Z_k \cdot e^{-i\pi\hat{\nu}_{kl}/4} \cdot C_j Z_k \cdot C_j S_k^\dagger, \qquad (157)$$

and the $C_j S_k^\dagger$ can be absorbed into the second $C_j Z_k$, yielding a sequence of four two-qubit gates, as desired. The four-qubit CPHASE can be implemented in place via fourteen two-qubit gates as follows. Note that for

$$\hat{\omega}_{jk} := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \qquad (158)$$

we have

$$C_j Z_k = e^{i\pi\hat{\omega}_{jk}/2}S_j^\dagger, \qquad (159)$$

and

$$\hat{\omega}_{jk} X_j = -X_j \hat{\omega}_{jk}. \qquad (160)$$

Hence,

$$\begin{aligned} & CCCZ_{jklm} \\ & = e^{i\pi\hat{\omega}_{lm}/4} \cdot \mathrm{Toffoli}_{jkl} \cdot e^{-i\pi\hat{\omega}_{lm}/4} \cdot \mathrm{Toffoli}_{jkl} \cdot CCS_{jkl}^\dagger, \end{aligned} \qquad (161)$$

where the Toffolis can be implemented via (157) and the $CCS^\dagger$ can be implemented similarly, so that the total cost for the four-qubit CPHASE is fourteen two-qubit gates. Hence the total cost of R is twice the cost of G, plus $4n$ Toffoli gates and $4n$ CNOTs, plus the four-qubit CPHASE, which comes to a total of

$$\begin{aligned} 8n^2 + 14 & \quad \text{two-qubit,} \\ 4 & \quad \text{single-qubit} \end{aligned} \qquad (162)$$

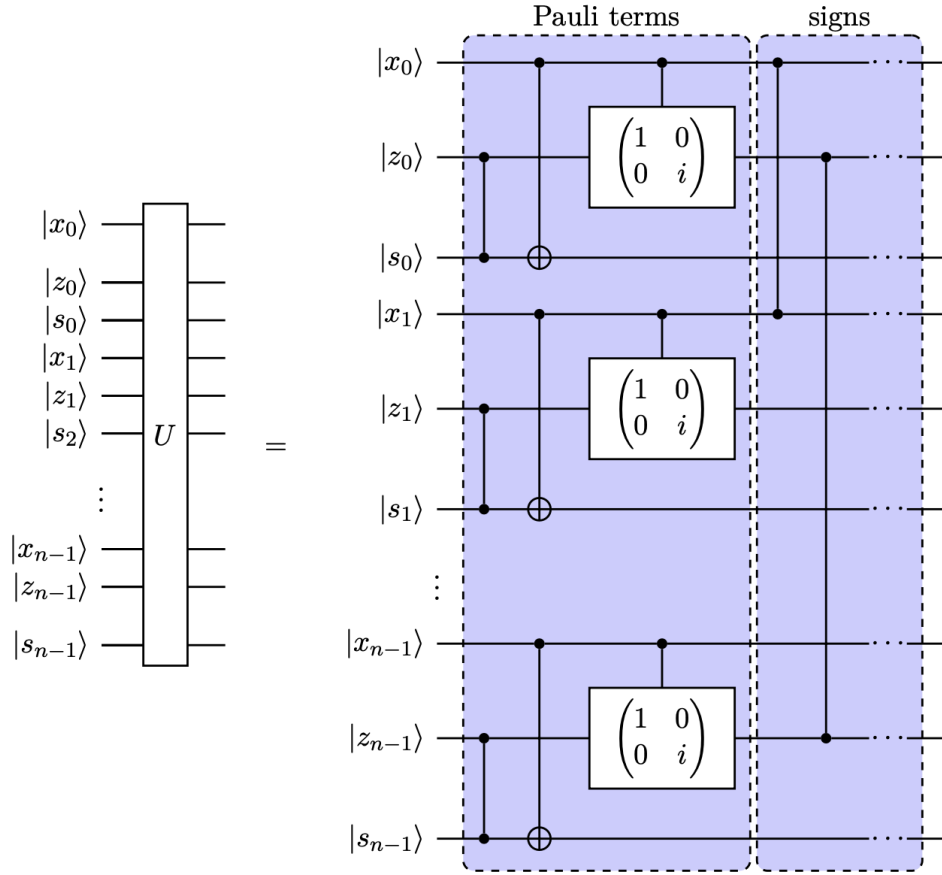gates. Implementing R also requires six additional auxiliary qubits in total.

Figure 5: Circuit diagram for U. The registers are $|\vec{x}\rangle$ and $|\vec{z}\rangle$, which contain the binary encodings of the Pauli terms, and $|s\rangle$, the system register. The first blue block, labeled 'Pauli terms,' implements the terms in the Hamiltonian controlled on the auxiliary qubits, with no signs. The second blue block then adds the signs by applying CZ between each pair of auxiliary qubits associated to a Pauli term with a negative sign. The two gates shown in this block are only examples; this block will take different forms for different Hamiltonians. The two examples shown would correspond to adding a negative sign to the terms $X_0 X_1$ (the first CZ in the 'signs' block) and $Z_0 Z_{n-1}$ (the second CZ in the 'signs' block).

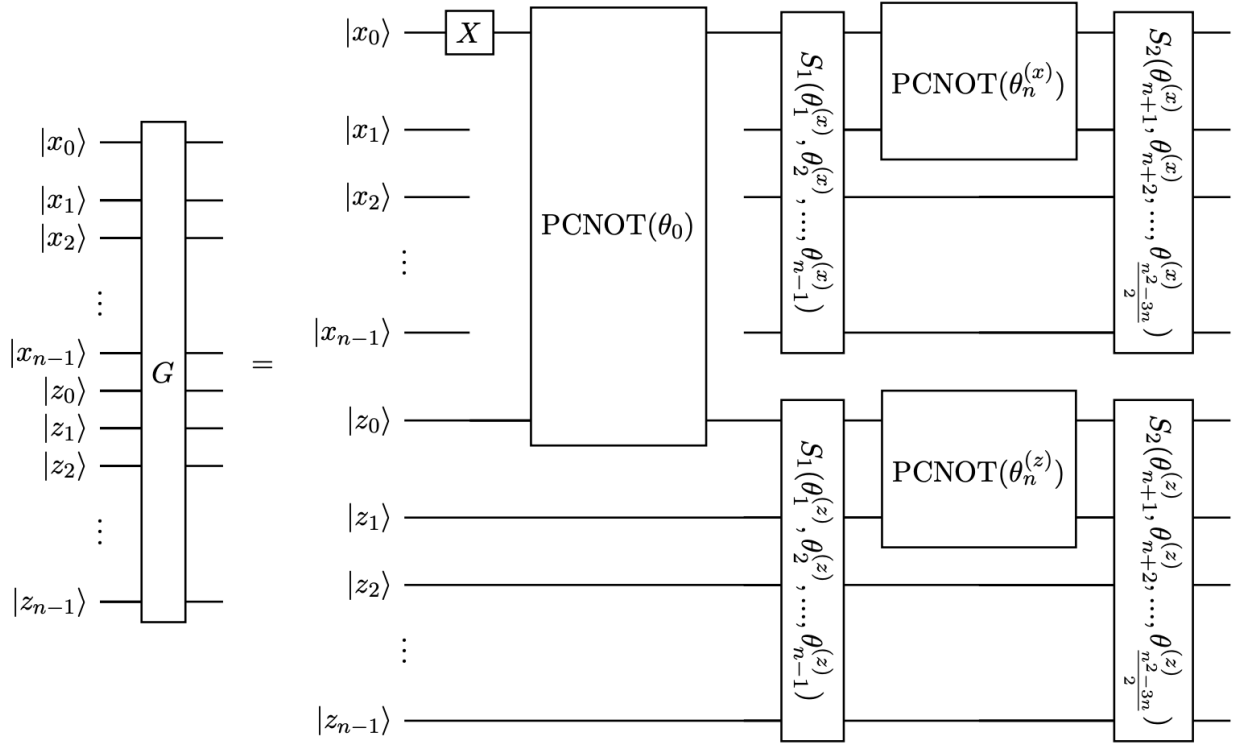Accepted in ⟨ ᐰuantum 2023-04-22, click title to verify. Published under CC-BY 4.0.

30

Figure 6: Circuit diagram for G. The gates labeled $S_1(\theta_1, \theta_2, ..., \theta_{n-1})$ and $S_2(\theta_{n+1}, \theta_{n+2}, ..., \theta_{\frac{n^2-3n}{2}})$ distribute the amplitudes of the weight-one and weight-two basis states in the superposition, and are defined in Fig. 7 and Fig. 8, respectively.
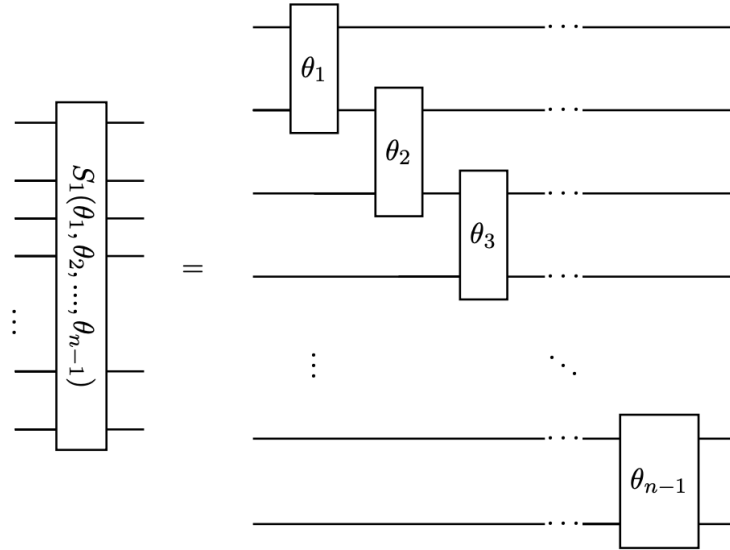

Figure 7: Circuit diagram for the $S_1(\theta_1, \theta_2, ..., \theta_{n-1})$ gate in Fig. 6. The gates labeled θ_i are partial swaps with gate angle θ_i.
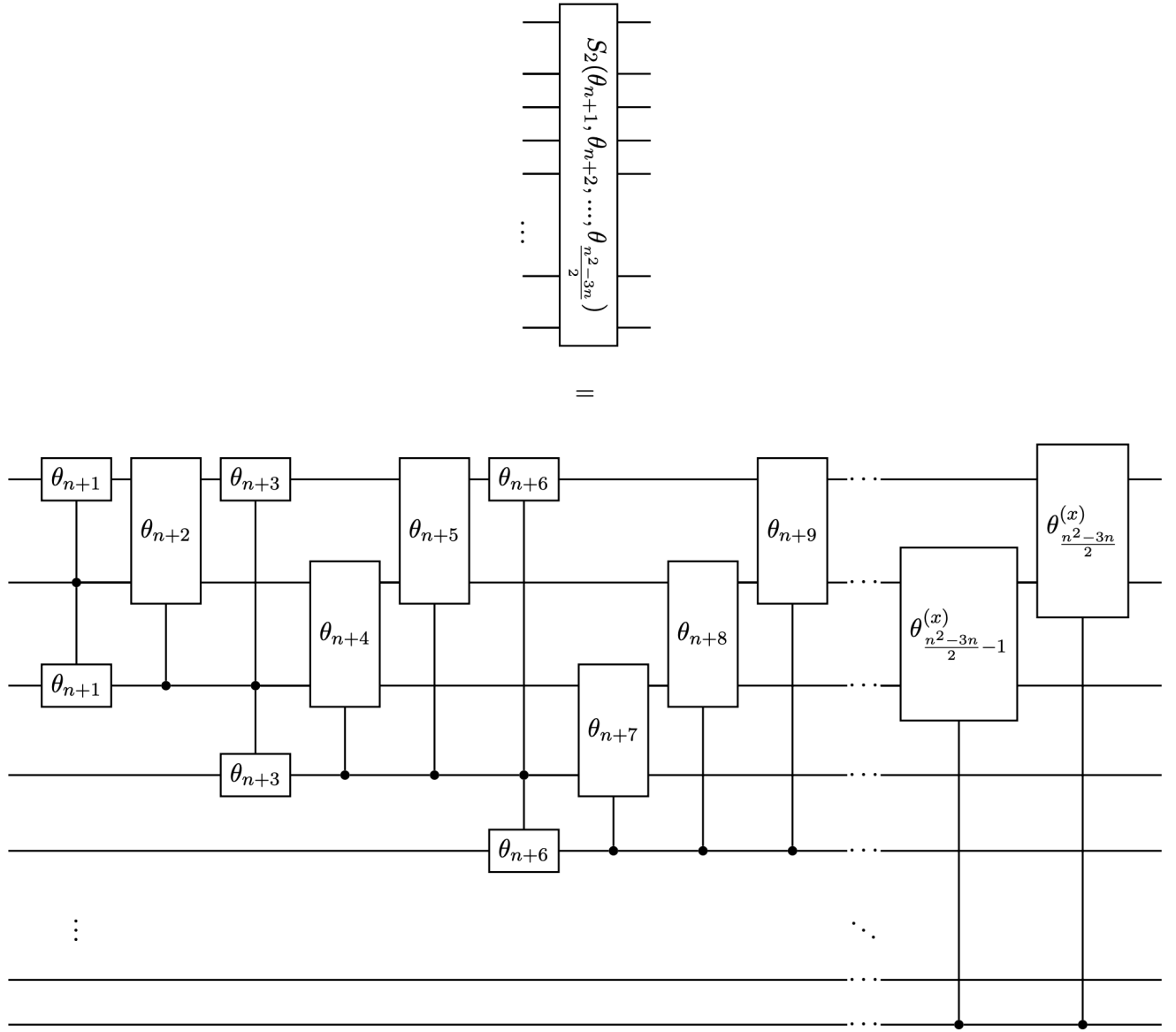
Figure 8: Circuit diagram for the $S_2(\theta_{n+1}, \theta_{n+2}, ..., \theta_{\frac{n^2-3n}{2}})$ gate in Fig. 6. The gates labeled θ_i are partial swaps with gate angle θ_i.
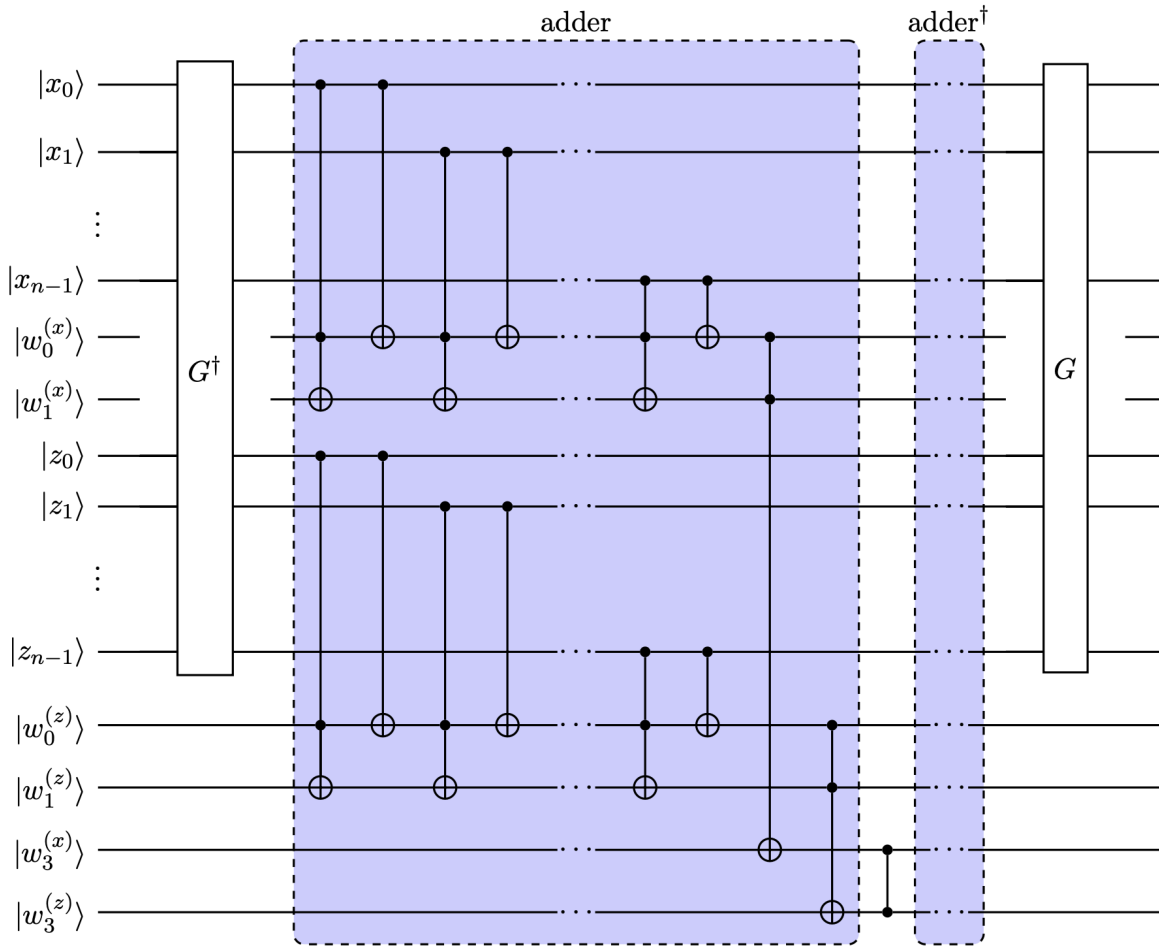
Figure 9: Circuit diagram for R. The blue block labeled 'adder' calculates the Hamming weight of each register $|\vec{x}\rangle$ and $|\vec{z}\rangle$, storing the results in $|w_1^{(x)}w_0^{(x)}\rangle$ and $|w_1^{(z)}w_0^{(z)}\rangle$, respectively, then calculates whether these weights are nonzero, storing the results in $|w_3^{(x)}\rangle$ and $|w_3^{(z)}\rangle$, respectively. The blue block labeled 'adder†' is the inverse of the 'adder' block. G is given in Fig. 6.