Tight bound for estimating expectation values from a system of linear equations

Abhijeet Alase, Robert R. Nerem, Mohsen Bagherimehrab, Peter Høyer, and Barry C. Sanders
Institute for Quantum Science and Technology, University of Calgary, 2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

(Received 9 December 2021; accepted 17 May 2022; published 24 June 2022)

The system of linear equations problem (SLEP) is specified by a complex invertible matrix A, the condition number κ of A, a vector $\boldsymbol{b}$, a Hermitian matrix M, and an accuracy ϵ, and the task is to estimate $\boldsymbol{x}^\dagger M\boldsymbol{x}$, where $\boldsymbol{x}$ is the solution vector to the equation $A\boldsymbol{x} = \boldsymbol{b}$. We aim to establish a lower bound on the complexity of the end-to-end quantum algorithms for SLEP with respect to ϵ, and devise a quantum algorithm that saturates this bound. To make lower bounds attainable, we consider query complexity in the setting in which a block encoding of M is given, i.e., a unitary black box U_M that contains M/α as a block for some $\alpha \in \mathbb{R}^+$. We show, by constructing a quantum algorithm and deriving a lower bound, that the quantum query complexity for SLEP in this setting is $\Theta(\alpha/\epsilon)$. Our lower bound is established by reducing the problem of estimating the mean of a black box function to SLEP. Our $\Theta(\alpha/\epsilon)$ result tightens and proves the common assertion of polynomial accuracy dependence [poly$(1/\epsilon)$] for SLEP without making any complexity-theoretic assumptions, and shows that improvement beyond linear dependence on accuracy is not possible if M is provided via block encoding.

DOI: 10.1103/PhysRevResearch.4.023237

I. INTRODUCTION

Systems of linear equations are essential to nearly every area of science, engineering, and mathematics, being particularly valuable in numerical solutions to differential equations and machine-learning applications [1–5]. For the system of linear equations problem (SLEP), a size N, an $N \times N$ complex invertible matrix A, an upper bound κ on the condition number of A, a vector $\boldsymbol{b}$, a Hermitian matrix M, and an accuracy ϵ are given, and the task is to estimate $\boldsymbol{x}^\dagger M\boldsymbol{x}$, where $\boldsymbol{x}$ is the complex solution vector to the equation $A\boldsymbol{x} = \boldsymbol{b}$. If the number of equations N is large, such as for big-data applications, solving SLEP is intractable for the best-known classical algorithms, which require polynomial resources in N. The landmark quantum algorithm by Harrow, Hassidim, and Lloyd (HHL algorithm) [6], together with its improvements [7–9], generate, to accuracy ϵ, a quantum state $|\boldsymbol{x}\rangle$ encoding the solution $\boldsymbol{x}$ using time and space resources that are polylogarithmic in N and $1/\epsilon$, which is denoted by polylog$(N, 1/\epsilon)$. SLEP can be solved by estimating the expectation value $\boldsymbol{x}^\dagger M\boldsymbol{x}$ using such quantum algorithms for generating $|\boldsymbol{x}\rangle$. Although it is commonly asserted that estimating $\boldsymbol{x}^\dagger M\boldsymbol{x}$ requires poly$(1/\epsilon)$ resources [10–12], this assertion is proven only under the assumption that A is given by oracles [6]. To this end, we establish a lower bound for the complexity of SLEP with respect to ϵ if M (instead of A) is given by an oracle, and we saturate this bound by constructing an end-to-end quantum algorithm for SLEP.

We now summarize the state of the art for the hardness of SLEP in terms of ϵ. Hardness results were proved for a restricted version of SLEP in which b is fixed to be a vector with a 1 in its first entry and zeros elsewhere, and M is fixed to be the diagonal matrix containing a 1 in the first $N/2 = 2^{n-1}$ diagonal entries and zeros in all other diagonal entries [6]. We refer to this restriction of SLEP with fixed M and b as F-SLEP (see Table I for a list of problems considered in this paper). Under a reasonable complexity-theoretic assumption (BQP $\neq$ PP), a polylog$(1/\epsilon)$ dependence was ruled out for any quantum algorithm for F-SLEP that achieves poly$(\log N, \kappa)$ dependence. Furthermore, any quantum algorithm solving F-SLEP with poly$(\log N, \kappa)$ dependence cannot achieve runtime faster than $O(1/\epsilon)$ if A is provided by an oracle. As F-SLEP is a restriction of SLEP, these hardness results extend to SLEP as well. However, complexity lower bounds that do not rely on complexity-theoretic assumptions are not known in the case that A is fixed. Moreover, whether a quantum algorithm with subpolynomial accuracy dependence can solve SLEP is not known in this case.

We next discuss previously developed end-to-end quantum algorithms for SLEP, which mirror the literature on SLEP lower bounds in their focus on restricted cases. For F-SLEP, M is the matrix representation of the observable $|0\rangle\langle 0| \otimes \mathbb{1}_{n-1}$ in the computational basis. Therefore, the expectation value $\boldsymbol{x}^\dagger M\boldsymbol{x}$ can be computed by repeatedly preparing the solution state $|\boldsymbol{x}\rangle$ and measuring the first qubit in the computational basis, then recording the fraction of measurements that yield a 0 outcome [6]. Efficient quantum algorithms were developed for estimating $|x_i|^2$ or $|x_i - x_j|^2$ for some indices i, j, with x_i the ith entry of $\boldsymbol{x}$ [13]. These cases, which are solved using amplitude estimation, are restrictions of SLEP to instances where entries of M are zero except for a single 1 on the diagonal, or except for a single 1 and a single -1 on the diagonal, respectively. In a quantum algorithm developed for

TABLE I. Abbreviation, name, and defining feature of all restrictions of SLEP and relevant computational problems considered in this paper.

Abbreviation	Problem name	Number	Comment
B-SLEP	Block-access system of linear equations problem	5	SLEP with block access to A, M
F-SLEP	Fixed System of Linear Equations Problem	2	SLEP in sparse-access setting with the restriction $M = \|0\rangle\langle 0\| \otimes \mathbb{1}_{n-1}$
S-EVHM	Sparse-access expectation value of a Hermitian matrix	6	Expectation value of a sparse M
B-EVHM	Block-access expectation value of a Hermitian matrix	4	Expectation value of a block encoded M
S-QLSP	Sparse-access quantum linear system problem	1	Output is an approximation of $\|\boldsymbol{x}\rangle$

data fitting, fit accuracy is computed by solving SLEP in the case $M = A$, i.e., the case that $\langle\boldsymbol{x}|M|\boldsymbol{x}\rangle \propto |\boldsymbol{b}\cdot\boldsymbol{x}|$ [1], where the expectation value is computed by the swap test [14]. As these algorithms apply only to a few restrictions of SLEP, computing expectation values in general could still be prohibitively expensive [15]; therefore, unclear complexities for general expectation-value estimation obfuscate potential applications of quantum algorithms for SLEP.

We now discuss our claims and the setting in which they are developed. We construct our lower bound and quantum algorithm in the block-encoding framework which provides an alternative over the sparse-access model in which matrices are provided by oracles for locating and evaluating their nonzero entries [16–21]. A block encoding U_H of a Hermitian matrix H is a unitary black box U_H that contains H/α_H in the upper-left block for some $\alpha_H \in \mathbb{R}^+$ [11,22,23]. We consider SLEP in the setting in which M is provided by block encoding U_M, a version of SLEP we refer to as B-SLEP. We show that any quantum algorithm for B-SLEP must make $\Omega(\alpha_M/\epsilon)$ queries to U_M. We then construct a quantum algorithm that saturates this bound, thereby showing that our lower bound for B-SLEP is tight, meaning that B-SLEP has query complexity $\Theta(\alpha_M/\epsilon)$. En route to deriving our lower bound for B-SLEP, we evaluate the complexity of estimating the expectation value of a block-encoded matrix M with respect to some given quantum state. In particular, we show that the query complexity of this problem is also in $\Theta(\alpha_M/\epsilon)$. We also derive a query-complexity lower bound for the restriction of SLEP to the case in which $A = \mathbb{1}_n$ and M is provided via sparse access. This restriction is equivalent to the problem of estimating the expectation value of a matrix M in the sparse-access setting. Our lower bound for this problem is $\Omega(d_M\beta_M/\epsilon)$, where the sparsity d_M of M and a bound β_M on the max-norm of M are supplied as inputs to the problem. We show that this lower bound is tight by constructing an explicit quantum algorithm.

Methodologically, we utilize the block-encoding framework to limit restrictions on M and to make lower bounds attainable. The key step for deriving our lower bound is reducing the problem of computing the mean of a black-box function to the problem of computing an expectation value of a block-encoded Hermitian matrix (B-EVHM). This lower bound is extended to B-SLEP by showing that B-EVHM reduces to B-SLEP. Our algorithm saturating this lower bound is constructed by combining a known algorithm for B-EVHM [24] with techniques for generating $|\boldsymbol{x}\rangle$.

Our complexity bound for B-SLEP makes rigorous the common assertion of poly$(1/\epsilon)$ dependence for SLEP and shows that improvement beyond linear-accuracy dependence is impossible if M is given by oracles. Our end-to-end quantum algorithm for solving systems of linear equations allows for a variable M as input, whereas previous algorithms consider a restricted M. Explicit complexities for SLEP are important to assessing if the measurement step is prohibitively resource expensive for a given application, thereby addressing the limitation raised by Aaronson's fourth caveat for quantum linear-equation solvers [15].

An outline of our work is as follows. In Sec. II we give background relevant to our results including a discussion of matrix encodings, quantum algorithms for linear algebra, and quantum algorithms for estimating expectation values. Next, in Sec. III, we describe our methodology and define B-SLEP rigorously. We present our lower bounds and quantum algorithms in Sec. IV and discuss these results in Sec. V. Finally, we give our conclusions in Sec. VI.

II. BACKGROUND

In this section, we first review the quantum linear systems problem, which is the computational problem of generating the solution state $|\boldsymbol{x}\rangle$ given A, $\boldsymbol{b}$, and κ, in the sparse-access setting. We then review the block-encoding framework of Hermitian matrices and relevant techniques for matrix arithmetic in this framework. Next, we state results relevant for deriving lower bound on accuracy dependence for the system of linear equations problem. Finally, we review known quantum algorithms for computing the expectation value of an unknown state in the context of system of linear equations or otherwise.

We start with some preliminaries. Denote the set of integers and positive integers by $\mathbb{Z}$ and $\mathbb{Z}^+$, respectively, and for $N \in \mathbb{Z}^+$ define

$$[N] := \{0, 1, \dots, N-1\}. \qquad (1)$$

Similarly, our convention for indexing rows and columns of a matrix is zero based. We denote the real and complex fields by $\mathbb{R}$ and $\mathbb{C}$, respectively, and the set of positive real numbers by $\mathbb{R}^+$. A number $\tilde{z}$ is an ϵ-additive approximation to another number z if $|z-\tilde{z}| < \epsilon$. Although we do not define a specific bit-string representation for real numbers and complex numbers, our results hold for any representation in which a number z has an r-bit representation encoding a number z' such that $|z-z'|/|z| \in O(2^{-r})$, and basic arithmetic operations on representations can be performed to r bits of precision in poly(r) time. These conditions hold for standard representations of real and complex numbers [25]. For a vector $|\psi\rangle \in \mathscr{H}$ in a finite-dimensional Hilbert space $\mathscr{H}$, we define $\|\,|\psi\rangle\,\| := \sqrt{\langle\psi|\psi\rangle}$, and for operators $V : \mathscr{H} \to \mathscr{H}$ define $\|V\|$ to be the corresponding operator norm. For a matrix $M \in \mathbb{C}^{N\times N}$, denote $\|M\|_{\max} := \max_{i,j\in[N]}\{|M_{ij}|\}$ as the max norm. For any $n \in \mathbb{Z}^+$ we define $\mathbb{1}_n$ to be the n-qubit identity operator. If n is clear from context, we may omit the subscript.

A. Quantum linear system problem in the sparse-access setting

We now explain the sparse-access model for access to Hermitian matrices. Next, we review a formulation of the quantum linear systems problem based on this model.

1. *Sparse access to Hermitian matrices*

The quantum linear systems problem [6,8] accepts an $N \times N$ matrix A and an N-dimensional vector $\boldsymbol{b}$ as inputs. To devise a quantum algorithm for generating the solution state in polylog(N) time, A and $\boldsymbol{b}$ must be encoded in such a way that they can be accessed efficiently. Most notably, such an efficiency requirement rules out encoding these inputs into a string containing a list of entries.

The standard formulation of the quantum linear systems problem utilizes unitary black boxes to encode A and $\boldsymbol{b}$ [6,8]. The vector $\boldsymbol{b}$ is given by a black box $U_{\boldsymbol{b}} \in \mathcal{U}(\mathscr{H}_2^{\otimes n})$ (n-qubit unitary operator) such that $U_{\boldsymbol{b}}|0^n\rangle = |\boldsymbol{b}\rangle$, where $|\boldsymbol{b}\rangle := \sum_i b_i |i\rangle / \| \sum_i b_i |i\rangle \|$ and $\{|i\rangle\}$ are computational basis states. The matrix A is described by sparse-access oracles, i.e., oracles for functions that provide the column index and the value of the nonzero entries of the matrix A in each given row [16]. We next explain in detail the sparse-access setting that we use in this paper.

For any $n \in \mathbb{Z}^+$, an *n-qubit Hermitian matrix H* is a $2^n \times 2^n$ Hermitian matrix. An n-qubit Hermitian matrix H is d_H *sparse* for $d_H \in [2^n]$ if each row of H has at most d_H nonzero entries. We call d_H the *sparsity parameter* of H. In the sparse-access setting, the matrix values $\{H_{jk}\}$ are encoded into both standard unitary oracles [21]

$$O_{H_{\text{val}}}|j\rangle|k\rangle|z\rangle = |j\rangle|k\rangle|z \oplus H_{jk}\rangle, \tag{2}$$

$$O_{H_{\text{loc}}}|j\rangle|l\rangle = |j\rangle|H_{\text{loc}}(j,l)\rangle, \tag{3}$$

where the function H_{loc} maps a row index j and a number $l \in [d_H]$ to the column index $H_{\text{loc}}(j,l)$ of the lth nonzero element in row j. Observe that the oracle $O_{H_{\text{loc}}}$ computes the location of entries in place, i.e., in the same quantum register as l. The oracle $O_{H_{\text{val}}}$ accepts a row j and column k index and returns the value H_{jk} in some binary format. The pair of oracles $O_{H_{\text{val}}}$ and $O_{H_{\text{loc}}}$ are a sparse-access encoding of H. For convenience, we make the following definition.

Definition 1. Sparse access to a Hermitian matrix H is a 4-tuple $(d_H, \beta_H, O_{H_{\text{val}}}, O_{H_{\text{loc}}})$, where $\beta_H \geqslant \|H\|_{\max}$ is an upper bound on the max-norm of H, d_H is the sparsity parameter of H, and where $O_{H_{\text{val}}}$ and $O_{H_{\text{loc}}}$ are unitary black boxes that form a sparse-access encoding of H.
We frequently turn to this definition in the description of inputs for computational problems in the sparse-access model.

We always assume that, if any unitary black box $O \in \mathcal{U}(\mathscr{H}_2^{\otimes n})$ is given, then unitary black boxes $O^\dagger$, c-O, and c-$O^\dagger$ are also given where c-$O := |0\rangle\langle 0| \otimes \mathbb{1}_n + |1\rangle\langle 1| \otimes O$. Consequentially, if we are reporting complexities, we equate the cost of a query to $O^\dagger$, c-O, and c-$O^\dagger$ to the cost of single query to O. The motivation for this assumption is that oracles model subroutines that have unknown internal structure and, given a description of a circuit for O, circuits for each $O^\dagger$, c-O, and c-$O^\dagger$ can always be constructed using a procedure independent of the specific circuit for O [26]. In particular, the circuit for c-O is found by replacing each gate G in the circuit for O with c-G. The circuit for $O^\dagger$ is found by reversing the circuit for O and replacing each gate G with its inverse $G^\dagger$. Finally, the circuit for c-$O^\dagger$ is found by replacing each gate G in the circuit for $O^\dagger$ with c-G.

2. *Complexity and limitations of the HHL algorithm*

We now state the quantum linear system problem rigorously. Next, we review the complexity of the HHL algorithm [6], which solves the quantum linear systems problem. We then discuss limitations and caveats of this quantum algorithm.

Given a system of linear equations

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{4}$$

define the solution state $|\boldsymbol{x}\rangle$ to be the quantum state $|\boldsymbol{x}\rangle := \sum_i x_i |i\rangle / \| \sum_i x_i |i\rangle \|$. The following problem statement is adapted from Ref. [8].

Problem 1 (Sparse-access quantum linear system problem (S-QLSP)). Given n qubits, a $\kappa \geqslant 1$, sparse access $(d_A, \beta_A, O_{A_{\text{val}}}, O_{A_{\text{loc}}})$ to a $2^n \times 2^n$ invertible Hermitian matrix A satisfying $\|A\| \leqslant 1$ and $\|A^{-1}\| \leqslant \kappa$, an accuracy $\epsilon \in (0, 2]$, and an n-qubit unitary black box $U_{\boldsymbol{b}}$, generate with probability at least $\frac{2}{3}$, and return "failure" otherwise, a state $|\tilde{\boldsymbol{x}}\rangle$ such that $\| |\tilde{\boldsymbol{x}}\rangle - |\boldsymbol{x}\rangle \| \leqslant \epsilon$ where $|\boldsymbol{x}\rangle$ is the solution state to the system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{b}$ is the 2^n-dimensional vector with entries $b_i = \langle i|U_{\boldsymbol{b}}|0^n\rangle$.

Tables II and III list input and output for the computational problems discussed in this paper. The crucial step in the HHL algorithm is a call to the phase-estimation algorithm, which in turn makes use of a quantum algorithm for sparse Hamiltonian simulation. The complexity of the HHL algorithm is poly($\log N, \kappa, 1/\epsilon$), where the poly($1/\epsilon$) dependence comes from the phase-estimation step. This complexity has been improved to poly($\log N, \kappa, \log(1/\epsilon)$) by replacing phase estimation with quantum-walk techniques for implementing A^{-1} [8]. These techniques have been generalized to the larger class of block-encoded matrices [22], which we discuss in the next section.

The HHL algorithm has four key limitations on its ability to solve linear equations in practice [15]. First, this algorithm only maintains an exponential speedup over classical techniques for instances of S-QLSP for which $\kappa \in O(\log(N))$. Next, the algorithm only applies to systems of linear equations in which the oracle $U_{\boldsymbol{b}}$ can be efficiently constructed. Similarly, the algorithm is only applicable if A can be encoded by sparse-access oracles, and the speedup is lost if $d_A \notin$ polylog(N). Finally, the output of HHL is a quantum state $|\boldsymbol{x}\rangle$ encoding the solution $\boldsymbol{x}$. As a result, the HHL algorithm is only useful for finding quantities that can be efficiently computed using $|\boldsymbol{x}\rangle$. The purpose of our work is to address this final caveat by evaluating the complexity of estimating the quantities of the form $\boldsymbol{x}^\dagger M \boldsymbol{x}$ given M.

TABLE II. Input and output for the computational problems listed in Table I. Here n is the number of qubits; κ is an upper bound on the condition number; A and M are Hermitian matrices of size $2^n \times 2^n$; $\boldsymbol{x}$ and $\boldsymbol{b}$ are vectors of length 2^n; α_A (α_M) is the scaling factor for the block encoding of A (M); a_A (a_M) is the number of ancilla qubits for block encoding of A (M); U_A (U_M) is the unitary block encoding of A (M); d_A (d_M) is the sparsity of A (M); β_A (β_M) is an upper bound on the max-norm of A (M); $O_{A_{\rm val}}$ ($O_{M_{\rm val}}$) is an oracle that returns the entry of A (M) in a given row and column; $O_{A_{\rm loc}}$ ($O_{M_{\rm loc}}$) is an oracle that returns the non-zero entries of A (M) in a given row; ϵ is an additive error tolerance; $U_{\rm b}$ is a black-box unitary, which generates the vector $\boldsymbol{b}$; V is another black-box unitary.

Abbreviation	Input	Output (approximation of)
B-SLEP	$(n, \kappa, (\alpha_A, a_A, 0, U_A), (\alpha_M, a_M, 0, U_M), \epsilon, U_{\boldsymbol{b}})$	$\boldsymbol{x}^\dagger M \boldsymbol{x}$
F-SLEP	$(n, \kappa, (d_A, \beta_A, O_{A_{\rm val}}, O_{A_{\rm loc}}), \epsilon, U_{\boldsymbol{b}})$	$\boldsymbol{x}^\dagger(\lvert 0\rangle\langle 0\rvert \otimes \mathbb{1}_{n-1})\boldsymbol{x}$
S-EVHM	$(n, (d_M, \beta_M, O_{M_{\rm val}}, O_{M_{\rm loc}}), \epsilon, V)$	$\langle 0^n \rvert V^\dagger M V \lvert 0^n\rangle$
B-EVHM	$(n, (\alpha_M, a_M, 0, U_M), \epsilon, V)$	$\langle 0^n \rvert V^\dagger M V \lvert 0^n\rangle$
S-QLSP	$(n, \kappa, (d_A, \beta_A, O_{A_{\rm val}}, O_{A_{\rm loc}}), \epsilon, U_{\boldsymbol{b}})$	$\lvert \boldsymbol{x}\rangle$

B. Block-encoding techniques for the quantum linear system problem

Quantum-walk techniques for implementing H^{-1} and Hamiltonian simulation (implementing e^{iH}) are applicable not only to sparse matrices, but to all matrices that can be encoded efficiently as blocks of larger unitary matrices. This realization has fueled development of quantum algorithms for linear algebra in which input matrices are provided through a block encoding [11,22,23]. We also derive our results in the block-encoding setting. In this section, we review block encoding of Hermitian matrices and describe techniques for matrix arithmetic that we use in the derivation of our results.

1. *Block encoding for Hermitian matrices*

Block encoding is a way of specifying a matrix input to a computational problem, and therefore provides an alternative to sparse-access encoding reviewed in Sec. II A 1. Intuitively, a block encoding of a Hermitian matrix H is a unitary matrix U_H which contains H as an upper-left block. We now adapt a rigorous definition of block encoding [22].

Definition 2. Let H be an n-qubit Hermitian matrix, $\alpha_H, \delta_H > 0$ and $a_H \in \mathbb{Z}^+$. A $U_H \in \mathcal{U}(\mathscr{H}_2^{\otimes(n+a_H)})$ is an $(\alpha_H, a_H, \delta_H)$-block encoding of H if

$$\| H - \alpha_H \langle 0^{a_H} \rvert U_H \lvert 0^{a_H}\rangle \| \leqslant \delta_H, \tag{5}$$

where $\langle 0^{a_H} \rvert U_H \lvert 0^{a_H}\rangle := \sum_{i,j\in[2^n]} (\langle 0^{a_H} \rvert \langle i \rvert U_H \lvert 0^{a_H}\rangle \lvert j\rangle) \lvert i\rangle\langle j\rvert$.

As α_H, a_H, and δ_H are provided along with U_H as inputs to a problem, we use the following convention.

Definition 3. Block access to a Hermitian matrix H is a tuple $(\alpha_H, a_H, \delta_H, U_H)$, where U_H is a unitary black box that is a $(\alpha_H, a_H, \delta_H)$ block encoding of H.

If block access $(\alpha_H, a_H, \delta_H, U_H)$ to H is given, we assume that $U_H^\dagger$, c-U_H, and c-$U_H^\dagger$ are also given. If a Hermitian matrix is given by sparse-access oracles, then a previously developed algorithm can be used to construct a block encoding for the same Hermitian matrix [22]. We now state a previously known lemma [22] that gives the query complexity of obtaining a block encoding from a sparse-access encoding. We adapt this lemma to include the complexity in terms of 2-qubit gates.

Lemma 1. Given a number of qubits $n \in \mathbb{Z}^+$ and sparse access $(d_H, \beta_H, O_{H_{\rm val}}, O_{H_{\rm loc}})$ to an n-qubit Hermitian matrix H, block access $(d_H\beta_H, 2, \epsilon, U_H)$ to H can be constructed such that U_H can be implemented with $O(1)$ queries to sparse-access oracles for H and $O(n + \log^{2.5}(\frac{d_H\beta_H}{\epsilon}))$ additional 2-qubit gates.

Proof. Define $\{\lvert\psi_j\rangle := \lvert 0\rangle \otimes \lvert j\rangle \otimes \lvert\varphi_j\rangle \in \mathscr{H}_2 \otimes \mathscr{H}_2^{\otimes n} \otimes \mathscr{H}_2^{\otimes n+1} : j \in [2^n]\}$, where

$$\lvert\varphi_j\rangle := \frac{1}{\sqrt{d_H}} \sum_{\substack{k=H_{\rm loc}(j,\ell)\\ \ell\in[d_H]}} \left(\sqrt{\frac{H_{jk}^*}{\beta_H}} \lvert 0\rangle \lvert k\rangle + \sqrt{1 - \frac{\lvert H_{jk}\rvert}{\beta_H}} \lvert 1\rangle \lvert k\rangle \right) \quad \forall\, j \in [2^n], \tag{6}$$

with $\sqrt{\cdot}$ denoting the principal square root.[1] Let $T \in \mathcal{U}(\mathscr{H}_2 \otimes \mathscr{H}_2^{\otimes n} \otimes \mathscr{H}_2^{\otimes(n+1)})$ be a unitary operator with action

$$T : \lvert 0^{n+2}\rangle \lvert j\rangle \mapsto \lvert\psi_j\rangle \quad \forall\, j \in [2^n], \tag{7}$$

which prepares the state $\lvert\psi_j\rangle$ conditional on the third register being in the state $\lvert j\rangle$. Let

$$R_T := 2TT^\dagger - \mathbb{1} = \sum_{j=1}^{2^n} \lvert j\rangle\langle j\rvert \otimes (2\lvert\varphi_j\rangle\langle\varphi_j\rvert - 1) \tag{8}$$

be the controlled Householder reflection [27], i.e., a reflection about $\lvert\varphi_j\rangle$ conditional on the state $\lvert j\rangle$ of the third register. Let $U_{\rm swap} \in \mathcal{U}(\mathscr{H}_2 \otimes \mathscr{H}_2^{\otimes n} \otimes \mathscr{H}_2^{\otimes n+1})$ be the operator that swaps the first $n+1$ qubits with the remaining $n+1$ qubits, respectively. Then a $(d_H\beta_H, n+2, \epsilon)$ block encoding of B is given by the unitary operator (Lemma 6, [22])

$$U_H = T^\dagger U_{\rm swap} R_T T. \tag{9}$$

We next explain how to implement this unitary operator.

A procedure for preparing $\lvert\varphi_j\rangle$ from the $\lvert 0\rangle$ state is key to performing T and R_T. A reflection about $\lvert\varphi_j\rangle$ is achieved by performing inverse state preparation, reflecting about $\lvert 0\rangle$, and then performing state preparation. Note that implementing T is equivalent to preparing the state $\lvert\varphi_j\rangle$ conditional on the third register being in the state $\lvert j\rangle$. Following the procedure given in the proof of Lemma 10 [21], the operator T can be implemented as follows. First, a Hadamard gate is performed on $\lceil \log d_H \rceil$ qubits, and then the oracle $H_{\rm loc}$ is applied to obtain

[1] Note that special care must be taken in choosing the sign of the square root if $H_{jk} \in \mathbb{R}^-$ [28].

the state

$$\frac{1}{\sqrt{d_H}}\sum_{\ell\in[d_H]}|0\rangle\,|H_{\mathrm{loc}}(j,l)\rangle\,. \tag{10}$$

The final step consists of using the oracle $O_{H_{\mathrm{val}}}$ and applying controlled rotation to the ancilla qubit to prepare the desired state. The unitary operators $T, T^\dagger$, and W can be implemented up to error at most ϵ in operator norm using $O(1)$ queries to the oracles encoding H and $O(n+\log^{2.5}(d_H\beta_H/\epsilon))$ additional 2-qubit gates [28]. As U_{swap} requires $O(n)$ 2-qubit gates, the complexities follow readily. ■

This lemma concludes our discussion of block access to a Hermitian matrix.

2. *Matrix arithmetic in the block-encoding setting*

We now review the matrix arithmetic techniques that we use for constructing a quantum algorithm for solving B-SLEP and for constructing our lower bound. Quantum algorithms for solving a system of linear equations can be reduced to using a block encoding of a matrix A to construct a block encoding of A^{-1}. The block encoding of A^{-1} can be obtained by an algorithm for quantum signal processing [22], which maps a block encoding of a given matrix H to that of $\mathcal{P}(H)$ for some given polynomial $\mathcal{P}$ of fixed degree and fixed coefficients. We also review a result that provides a polynomial for approximating the inversion function $\mathrm{inv}: x\mapsto 1/x$.

The complexity of quantum signal processing is given in the following theorem.

Theorem 1. (Reference [11]). Given block access $(\alpha_A, a_A, \delta_A, U_A)$ to a Hermitian matrix A of any size, an error $\sigma>0$ and a polynomial $\mathcal{P}:\mathbb{R}\to\mathbb{R}$ of degree d satisfying $|\mathcal{P}(x)|\leqslant\frac{1}{2}$ for all $x\in[-1,1]$, then block access $(1, a_A+2, 4d\sqrt{\delta_A/\alpha_A}+\sigma, U_{\mathcal{P}(A/\alpha_A)})$ to $\mathcal{P}(A/\alpha_A)$ can be constructed such that $U_{\mathcal{P}(A/\alpha_A)}$ makes $2d+1$ queries to U_A and $O(a_A d)$ additional 2-qubit gates.

The original proof of Theorem 1 constructs a circuit for implementing $U_{P(A/\alpha_A)}$. Circuits for $U^\dagger_{P(A/\alpha_A)}$, c-$U_{P(A/\alpha_A)}$, and c-$U^\dagger_{P(A/\alpha_A)}$ can be constructed using the procedure explained in the preliminaries. Note that c-c-U_A can be constructed using c-U_A and two Toffoli gates.

We use the following corollary about approximation of the function $\mathrm{inv}(x)$ with polynomials.

Corollary 1 (Reference [11]). For any $\epsilon,\delta\in(0,\frac{1}{2}]$, a polynomial $\mathcal{P}:\mathbb{R}\to\mathbb{R}$ with odd degree $\mathcal{O}(\frac{1}{\delta}\log(\frac{1}{\epsilon}))$ exists such that for all $x\in[-1,1]\setminus[-\delta,\delta]$ then $|\mathcal{P}(x)|\leqslant 1$ and

$$|\mathcal{P}(x)-f(x)|<\epsilon, \tag{11}$$

where $f(x)=\frac{3\delta}{4}\,\mathrm{inv}(x)$.

This corollary concludes our discussion of block-encoding techniques for matrix arithmetic.

C. Complexity of solving a system of linear equations

We now review the results that describe the hardness of solving restrictions of SLEP with respect to accuracy. We also state a previously known query-complexity lower bound for computing the mean of a black-box function that we later use to derive a new lower bound for SLEP.

1. *Previous results on the accuracy dependence of system of linear equations*

It is often asserted that quantum algorithms for solving SLEP must have $\mathrm{poly}(1/\epsilon)$ accuracy dependence due to the cost of estimating $\langle\boldsymbol{x}|\,M\,|\boldsymbol{x}\rangle$ [6,7,10–12]. However, this assertion is proven to hold only under certain restrictions. One of the existing results shows that $\mathrm{polylog}(1/\epsilon)$ dependence of the time complexity is not achievable under complexity-theoretic assumptions. Another previous hardness result shows that any quantum algorithm with $\mathrm{polylog}N$ dependence has runtime in $O(1/\epsilon)$; however, as this result is derived by bounding queries to oracles for A, it is not applicable if A is fixed. We now review these two results in detail.

We adopt the standard big-O definition for single variables and the following definition for multiple variables [29].

Definition 4. Let $f,g:\{\mathbb{R}^+\}^m\to\mathbb{R}^+$. Then $f(x_1,x_2,\dots,x_m)\in O(g(x_1,x_2,\dots,x_m))$ if there exist constants $\eta, C\in\mathbb{R}^+$ such that for all $x_1,x_2,\dots,x_m>\eta$,

$$f(x_1,x_2,\dots,x_m)\leqslant Cg(x_1,x_2,\dots,x_m). \tag{12}$$

We define multivariate Ω analogously.

We now rigorously define F-SLEP, a previously studied problem [6] which we first reviewed in Sec. I.

Problem 2 (Fixed M and b system of linear equations problem (F-SLEP)). Given $N\in\mathbb{Z}^+$, a $\kappa\geqslant 1$, sparse access $(d_A,\beta_A,O_{A_{\mathrm{val}}},O_{A_{\mathrm{loc}}})$ to an $N\times N$ matrix A with $\|A\|\leqslant 1$ and $\|A^{-1}\|\leqslant\kappa$, and an accuracy $\epsilon\in(0,1]$, return an ϵ-additive approximation of $\langle\boldsymbol{x}|M|\boldsymbol{x}\rangle$ with probability at least $\frac{2}{3}$, where $|\boldsymbol{x}\rangle$ is the unit vector proportional to $A^{-1}|\boldsymbol{0}\rangle$ and M is the diagonal matrix containing a 1 in the first $N/2$ diagonal entries and a zero in all other entries.

Two hardness results are known for F-SLEP [6]. The first result establishes a lower bound on the oracle queries to A, and its proof relies on a query-complexity lower bound for computing the parity of a Boolean expression.

Theorem 2 (Reference [6]). A quantum algorithm can solve F-SLEP using ν queries to oracles for A with $\nu\in N^{c_1}\mathrm{poly}(\kappa)/\epsilon^{c_2}$ only if $c_1+c_2\geqslant 1$.

The original statement of this theorem refers to the time complexity of quantum algorithms for SLEP rather than number of queries [6]. However, the proof given is for a stronger statement in terms of queries to oracles for A that we give in Theorem 2.

The next hardness result for F-SLEP is in terms of time complexity and relies on the complexity-theoretic assumption BQP $\neq$ PP.

Theorem 3 (Reference [6]). If BQP $\neq$ PP, then there does not exist a quantum algorithm that solves F-SLEP running in time $\mathrm{poly}(\kappa,\log(N),\log(1/\epsilon))$.

In fact, the proof of this theorem proves the complexity $\mathrm{polylog}(N,1/\epsilon)$ for the restriction $A=\mathbb{1}$, which is a stronger result. Note that BQP $=$ PP contradicts the widely held conjecture BQP $\neq$ NP.

2. *Query complexity of approximating the mean of a black-box function*

We now review a query-complexity lower bound for computing the approximate mean of a black-box function [30]. We use this lower bound to later derive a lower bound for B-SLEP

Algorithm 1. Amplitude-estimation algorithm.

```
Intput: A number of qubits $n \in \mathbb{Z}^+$, an accuracy $\epsilon \in (0,1)$, two $n$-qubit unitary black boxes $V, W$
Output: With probability at least $\frac{2}{3}$, an $\epsilon$-additive approximation $\tilde{r}$ to the amplitude $r = |\langle 0|W^\dagger V W|0\rangle|$
Procedure:
 1: Construct a quantum circuit for c-$S$ where $S = W P_0 W^\dagger V W P_0 W^\dagger V^\dagger$ and $P_0 = \mathbb{1} - 2|0^n\rangle\langle 0^n|$.
 2: Apply phase estimation with input $(n, 2\epsilon, \text{c-}S, W)$ to obtain with probability at least $\frac{2}{3}$ a $(2\epsilon)$-additive approximation $\tilde{\theta}$ of some
    eigenphase $\theta$ of $S$.
 3: return $\tilde{r} = |\cos(\tilde{\theta}/2)|$.
```

in Sec. IV A. As is standard, we say an oracle O_f encodes [31] a discrete function $f : \{0,1\}^{r_1} \to \{0,1\}^{r_2}$ for $r_1, r_2 \in \mathbb{Z}^+$ if

$$O_f : |j\rangle\,|s\rangle \mapsto |j\rangle\,|f(j) \oplus s\rangle\,, \quad \forall\, j \in \{0,1\}^{r_1}, s \in \{0,1\}^{r_2}, \tag{13}$$

where $\oplus$ denotes bitwise XOR operation.

We are now ready to state the approximate-mean (AM) problem.

Problem 3 (Approximate mean). Given $N \in \mathbb{Z}^+$ and $\epsilon \in (1/2N, 1]$ and given an oracle O_f encoding a function $f : [N] \mapsto [0,1]$, return, with probability at least $\frac{2}{3}$, an ϵ-additive approximation of $\mu_f := (\sum_{j=0}^{N-1} f(j))/N$.

A query-complexity lower bound for this problem is given by Nayak and Wu ([30], Corollary 1.12).

Corollary 2. The query complexity of the problem AM is $\Omega(1/\epsilon)$.

This concludes our review of relevant hardness results.

D. A reformulation of amplitude-estimation algorithm

We now review a reformulation [32] of the amplitude-estimation algorithm [33]. This reformulation is used in some quantum algorithms for expectation-value estimation that we review in this section. We begin by reviewing the complexity of the standard phase-estimation algorithm [31]. Next, we describe how phase estimation is used to construct the amplitude-estimation algorithm.

Phase estimation is a key subroutine in many quantum algorithms that achieve superpolynomial speedup over classical algorithms. The inputs, output, and complexity of the phase-estimation algorithm are given by the following theorem, which is an adaptation from the standard description of the algorithm [31].

Theorem 4 (Phase-estimation algorithm). There exists a quantum algorithm that accepts n-qubits, an accuracy ϵ, and n-qubit unitary black boxes V and W, that, with probability at least $\frac{2}{3}$, samples an ϵ-additive approximation to the eigenphase θ of V corresponding to the eigenvector $|\psi\rangle$ with probability $|\langle\psi|W|0\rangle|^2$. This algorithm makes $O(1/\epsilon)$ queries to V, a single query to W, and $O(\log^2(1/\epsilon))$ additional 2-qubit gates.

Now we show how phase estimation is used to perform amplitude estimation. In amplitude estimation [32], the inputs are n qubits, an accuracy $\epsilon \in (0,1)$, and two n-qubit unitary black boxes V and W. The output is an ϵ-additive approximation to the amplitude $r := |\langle 0|W^\dagger V W|0\rangle|$ with probability at least $\frac{2}{3}$. Define $|\psi_0\rangle := W\,|0\rangle$, $|\psi_1\rangle := V\,|\psi\rangle$, $P_0 := \mathbb{1} - 2\,|0^n\rangle\langle 0^n|$, and define two reflection operators $S_0 := \mathbb{1} - |\psi_0\rangle\langle\psi_0| = W P_0 W^\dagger$ and $S_1 := \mathbb{1} - |\psi_1\rangle\langle\psi_1| = V W P_0 W^\dagger V^\dagger$. The composition of these two reflection operators $S := S_0 S_1$ is a rotation operator in the two-dimensional space spanned by $|\psi_0\rangle$ and $|\psi_1\rangle$ that rotates $|\psi_0\rangle$ toward $|\psi_1\rangle$ by angle $2\theta := 4\arccos(r)$. This implies that the eigenvalues of S are $e^{i\theta}$, and the amplitude returned by amplitude estimation is $r = |\cos(\theta/2)|$. Therefore, phase estimation can be used to estimate r by estimating an eigenphase of the operator S. Note that this call to phase estimation uses c-S, which can be constructed by replacing each factor of V and W in the definition of S with c-V and c-W. The complexity of this algorithm is given in the following lemma, which follows readily from the analysis in Ref. [32].

Lemma 2. Algorithm 1 with input (ϵ, V, W) makes $O(\frac{1}{\epsilon})$ queries to V and W and uses $O(\frac{n}{\epsilon})$ additional 2-qubit gates.

Proof. The complexity for queries to V and W follows directly from the complexities in Theorem 4. Note that P_0 can be implemented using $O(n)$ 2-qubit gates ([31], p. 251) which means the call to the phase-estimation algorithm uses a total of $O(n/\epsilon)$ 2-qubit gates for queries to P_0 and $\log^2(1/\epsilon)$ 2-qubit gates for other operations. ■

This lemma concludes our review of amplitude estimation.

E. Quantum algorithms for computation of expectation value

Quantum algorithms for estimating expectation values have been developed both in the context of SLEP [1,6,13] and in a variety of other applications [24,32,34–36]. We now review quantum-algorithmic techniques for estimating expectation values and discuss limitations of these techniques.

1. *Estimating expectation values in restrictions of SLEP*

We begin by discussing techniques for estimating expectation values that occur in quantum algorithms for SLEP. We review three techniques. These techniques use single-qubit computational-basis measurements, the swap test, and phase estimation, respectively.

We now discuss the first technique. In F-SLEP (Problem 2), M is the matrix representation of the operator $|0\rangle\langle 0| \otimes \mathbb{1}_{n-1}$ which is the orthogonal projector on the first qubit. As a result, the expectation value of a state $|\boldsymbol{x}\rangle$ with respect to M can be estimated by repeatedly preparing $|\boldsymbol{x}\rangle$, measuring the first qubit in the computational basis, and averaging the outcomes [6]. This approach requires $O(1/\epsilon^2)$ preparations of $|\boldsymbol{x}\rangle$ to obtain an estimate of the expectation value up to additive error ϵ [6].

We now discuss the second technique. If $M = A^{-1}$, where A defines the system of linear Eq. (4), then a swap test can be used to estimate $\langle\boldsymbol{x}|M|\boldsymbol{x}\rangle = \langle\boldsymbol{x}|\boldsymbol{b}\rangle$, as was demonstrated in an application of the HHL algorithm to least-square fitting [1]. This estimation is achieved by first preparing the state $|+\rangle\,|\boldsymbol{b}\rangle\,|\boldsymbol{x}\rangle$ using an algorithm for QLSP, with $|+\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$, followed by a controlled-swap operation, with control on the first qubit, to obtain the state $(|0\rangle\,|\boldsymbol{b}\rangle\,|\boldsymbol{x}\rangle + |1\rangle\,|\boldsymbol{x}\rangle\,|\boldsymbol{b}\rangle)/\sqrt{2}$. Finally, the first (control) qubit is measured in the computational basis. The probability that this measurement returns the outcome 0 is $(1 + |\,\langle \boldsymbol{x}|\boldsymbol{b}\rangle\,|^2)/2$. To obtain an estimate of $|\,\langle \boldsymbol{x}|\boldsymbol{b}\rangle\,|^2$ to additive error ϵ, the swap test needs to be carried out $O(1/\epsilon^2)$ times.

If every entry of M is zero except for a single 1 on its ith diagonal entry, then the expectation value $\langle \boldsymbol{x}|M|\boldsymbol{x}\rangle = |\,\langle \boldsymbol{x}|i\rangle\,|^2$ can be estimated by a direct application of the amplitude-estimation algorithm [33]. This technique was used for constructing an algorithm for computing the effective resistance in a given electrical network [11,13]. The time complexity of this method scales as $O(1/\epsilon)$ with respect to the additive error ϵ.

The key limitation of all three techniques discussed above is that they each apply only to some limited class of matrices; the first technique applies if M represents the orthogonal projector on the first qubit, the second technique applies if $M = A^{-1}$, and the final technique applies if M has entries that are all zero except for a single 1 on the diagonal. Moreover, the first two techniques, which rely on classical sampling, achieve $O(1/\epsilon^2)$ rather than $O(1/\epsilon)$.

2. *Estimating expectation values of simulatable Hermitian matrices*

We now discuss techniques for estimating expectation values, all of which are outside of the context of solving SLEP. The four techniques rely on the Hadamard test and algorithms for Hamiltonian simulation, phase estimation, and amplitude estimation.

We now review the first technique, which is a quantum algorithm for estimating $\langle 0^n|V^\dagger U V|0^n\rangle$, where U and V are given n-qubit unitary black boxes. One common method for this estimation is the Hadamard test and proceeds as follows [37]. First prepare the state $|+\rangle\,|\psi\rangle$, and then apply the controlled unitary operator c-U (with control on the first qubit) to obtain the state $(|0\rangle\,|\psi\rangle + |1\rangle\,U\,|\psi\rangle)/\sqrt{2}$, where $|\psi\rangle := V\,|0\rangle$. Next, apply a Hadamard gate H on the first qubit and measure this qubit in the computational basis. The expectation value of the output is $\mathrm{Re}\,\langle\psi|U|\psi\rangle = \mathrm{Re}\,\langle 0^n|V^\dagger U V|0^n\rangle$. To obtain an ϵ-additive approximation of the real part of the expectation value, $O(1/\epsilon^2)$ runs of the Hadamard test are required. The imaginary part of the expectation value can be estimated similarly by instead starting with the state $(|0\rangle - i\,|1\rangle)\,|\psi\rangle\,/\sqrt{2}$.

The second technique for estimating the expectation value of U relies on amplitude estimation [32]. Reduction to amplitude estimation makes use of the relations

$$\begin{aligned}\mathrm{Re}(\langle\psi|U|\psi\rangle) &= (4|\langle +|\,\langle\psi|\,\text{c-}U\,|+\rangle\,|\psi\rangle|^2 \\ &\quad - |\langle\psi|U|\psi\rangle|^2 - 1)/2,\\ \mathrm{Im}(\langle\psi|U|\psi\rangle) &= (4|\,\langle +|\,\langle\psi|\,(e^{i\sigma_z\pi/4}\otimes \mathbb{1}_n)\text{c-}U\,|+\rangle\,|\psi\rangle\,|^2 \\ &\quad - |\,\langle\psi|U|\psi\rangle\,|^2 - 1)/2. \qquad (14)\end{aligned}$$

The terms $|\,\langle +|\,\langle\psi|\,\text{c-}U\,|+\rangle\,|\psi\rangle\,|^2$, $|\,\langle +|\,\langle\psi|\,(e^{i\sigma_z\pi/4}\otimes \mathbb{1}_n)\text{c-}U\,|+\rangle\,|\psi\rangle\,|^2$, and $|\,\langle\psi|U|\psi\rangle\,|^2$ on the right-hand sides of Eq. (14) can each be estimated by making one call each to the amplitude-estimation algorithm. We review amplitude estimation in Sec. II D. This approach yields an algorithm for computing the expectation value $\langle\psi|U|\psi\rangle$ with complexity $O(1/\epsilon)$.

Whereas the two techniques above apply only to estimating the expectation value of unitary matrices, they are used as subroutines in other techniques for estimating expectation values of Hermitian matrices. We now review a technique for estimating the expectation value of any simulatable matrix M [32]. A $N \times N$ matrix M is simulatable if, for any $t \in \mathbb{R}^+$ and $\epsilon \in (0,1)$, the propagator e^{iMt} can be implemented by a poly$(\log(N), t, 1/\epsilon)$-sized circuit [16]. The expectation value $\langle\psi|M|\psi\rangle$ is then obtained by using the approximation [32]

$$|\,\langle\psi|e^{-iMt}|\psi\rangle - 1 + it\,\langle\psi|M|\psi\rangle\,| \in O(t^2). \qquad (15)$$

The term $\langle\psi|e^{-iMt}|\psi\rangle$ is computed by using one of the techniques for estimating the expectation value of a unitary matrix, which we discussed above. If M is simulatable by a poly$(n, t, \log(1/\epsilon))$-sized circuit, then $\langle\psi|M|\psi\rangle$ can be computed to additive approximation ϵ in $\widetilde{O}((1/\epsilon)^{1+\alpha})$ time for arbitrarily small α using an improved version of this technique. Here $\widetilde{O}$ is called soft O and indicates suppression of logarithmic factors.

An alternative technique for estimating the expectation value of a Hermitian and simulatable M employs the phase-estimation algorithm applied to the operator e^{iMt} [35]. Suppose M has spectral decomposition $M = \sum_j \lambda_j\,|v_j\rangle\,\langle v_j|$, and suppose $|\psi\rangle$ has decomposition $|\psi\rangle = \sum_j \alpha_j\,|v_j\rangle$. One application of the phase-estimation circuit results in a state close to $|\psi_M\rangle = \sum_j \alpha_j\,|v_j\rangle\,|\lambda_j\rangle$. Measuring the final register in the computational basis yields λ_j with probability approximately $|\alpha_j|^2$ for each j. The expectation value $\langle\psi|M|\psi\rangle = \sum_j |\alpha_j|^2\lambda_j$ can then be estimated by repeating the above procedure $O(1/\epsilon^2)$ times and averaging the measurement outcomes. Assuming that M is simulatable by a poly$(n, t, \log(1/\epsilon))$-sized circuit, the total complexity of this algorithm is evidently $\widetilde{O}(1/\epsilon^3)$, which is not explicitly stated.

In conclusion, the first two techniques are particularly relevant to our work only for building the latter two techniques. These final two techniques for estimating the expectation value outside of the context of SLEP apply to simulatable Hermitian matrices, which include all sparse matrices [16]. The best of these techniques achieves close to linear dependence on $1/\epsilon$. These techniques have not been combined with the algorithms for QLSP to solve SLEP.

3. *Estimating expectation values of block-encoded Hermitian matrices*

In this section, we review an algorithm for estimating the expectation value of a block-encoded Hermitian matrix with respect to a quantum state given by a unitary black box [24]. The algorithms developed for estimating the expectation value of simulatable Hermitian matrices can also be used for estimating the expectation values of block-encoded matrices. However, there is an elegant and more efficient way to estimate the expectation value of block-encoded matrices.

We first adapt the following problem from Lemma 5 in Ref. [24].

Algorithm 2. Algorithm for block-access expectation value of an Hermitian matrix.

```
Input: $n$ qubits, block access $(\alpha_M, a_M, 0, U_M)$ to $M$, an accuracy $\epsilon > 0$, and a unitary black box $V$
Output: an $\epsilon$-additive approximation $\tilde{u}$ to $u := \langle 0|V^\dagger M V|\psi\rangle$ with probability $\frac{2}{3}$
Procedure:
    Construct block access $(1, a_M, 0, U_{M'})$ to $M' = \text{c-}(M/\alpha_M)$        ▷ use Lemma 4.
    Compute $\tilde{r}$ by executing Algorithm 1 with input $(n + a_M, \epsilon/2\alpha_M, U_{M'}, \mathbb{1}_{a_M} \otimes H \otimes V)$.
    return $\tilde{u} := \alpha_M(2\tilde{r} - 1)$.
```

Problem 4(Block-access expectation value of a Hermitian matrix (B-EVHM)). Given n qubits, block access $(\alpha_M, a_M, 0, U_M)$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon \in [\alpha_M/2^n, \alpha_M]$, and an n-qubit unitary black box V, return with probability at least $\frac{2}{3}$ an ϵ-additive approximation to $\langle 0^n|V^\dagger M V|0^n\rangle$.

We now review an algorithm for B-EVHM given in Ref. [24] that achieves linear dependence on $1/\epsilon$. We include a full analysis of the complexity of this algorithm by counting queries to V and U_M, and additional two-qubit gates separately. To simplify this analysis, we employ the standard amplitude-estimation algorithm, as opposed to the more recent amplitude-estimation algorithms which do not require the quantum Fourier transform [38]. This change in amplitude-estimation algorithm has no effect on complexity.

As the expectation value of M is not always positive, it cannot be computed by a direct application of amplitude estimation. The problem is first reduced to one solvable by amplitude estimation in the following lemma, using a technique that employs a controlled block encoding [32].

Lemma 3. For any n-qubit Hermitian matrix M with $\|M\| \leqslant \Lambda$ and for any n-qubit state $|\psi\rangle$,

$$\langle\psi|M|\psi\rangle = \Lambda[2|\langle +|\,\langle\psi|\,\text{c-}(M/\Lambda)\,|+\rangle\,|\psi\rangle| - 1]. \tag{16}$$

Proof. The matrix M is Hermitian; therefore, $\langle\psi|M|\psi\rangle \in \mathbb{R}$ for any $|\psi\rangle \in \mathscr{H}_2^{\otimes n}$. As $\|M/\Lambda\| \leqslant 1$, we have $|\langle\psi|M/\Lambda|\psi\rangle| \leqslant 1$ and $\langle\psi|M/\Lambda|\psi\rangle + 1 = |1 + \langle\psi|M/\Lambda|\psi\rangle|$. Equation (16) follows from

$$\begin{aligned} &2|\langle +|\langle\psi|\text{c-}(M/\Lambda)|+\rangle|\psi\rangle| \\ &\quad = 2|\langle +|\langle\psi|(|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes M/\Lambda)|+\rangle|\psi\rangle| \\ &\quad = |1 + \langle\psi|M/\Lambda|\psi\rangle| \\ &\quad = 1 + \langle\psi|M/\Lambda|\psi\rangle. \end{aligned} \tag{17}$$

■

From the definition of block encoding (Definition 2) it is evident that $\|M\| \leqslant \alpha_M$ for any block encoding. For our application of Lemma 3, α_M, which is included in block access to M, is used as the bound Λ on the norm of M. The next lemma, which is a special case of Lemma 52 in Ref. [11], shows that block access to c-(M/α_M) can be implemented if block access to M is given.

Lemma 4 (Reference [11]). Given n qubits and block access $(\alpha_M, a_M, 0, U_M)$ to an $2^n \times 2^n$ Hermitian matrix M. Then block access $(1, a_M, 0, U_{M'})$ to the Hermitian matrix $M' :=$ c-(M/α_M) can be implemented such that $U_{M'}$ uses a single query to U_M.

Proof. The operator U_M is an $(n + a_M)$-qubit unitary. The desired circuit for $U_{M'}$ acts on $a_M + 1 + n$ qubits and comprises only a single query to c-U_M in which the $(a_M + 1)$th qubit is the control qubit, and the U_M operation controlled by this qubit operates on the first a_M qubits and the last n qubits. We now show that the operator $U_{M'}$ implemented by this circuit is indeed a $(1, a_M, 0)$ block encoding of c-(M/α_M). Let s_1 and s_2 be arbitrary n-bit strings. Then

$$\langle 0^{a_M}, 0, s_1|U_{M'}|0^{a_M}, 0, s_2\rangle = \delta_{s_1,s_2}, \tag{18}$$

as the control qubit is set to $|0\rangle$, so U_M is not applied. Furthermore,

$$\langle 0^{a_M}, 1, s_1|U_{M'}|0^{a_M}, 1, s_2\rangle = M_{s_1,s_2}/\alpha_M, \tag{19}$$

by definition of block encoding. Finally, $\langle 0^{a_M}, 0, s_1|U_{M'}|0^{a_M}, 1, s_2\rangle = 0$. By Eqs. (18) and (19),

$$\langle 0^{a_M}|U_{M'}|0^{a_M}\rangle = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & M/\alpha \end{pmatrix}, \tag{20}$$

where 0 is a $2^n \times 2^n$ zero matrix. As Eq. (20) gives the matrix representation of c-(M/α), the operator $U_{M'}$ is the desired block encoding. ■

The following theorem states the complexity of solving B-EVHM.

Theorem 5. Given n qubits, block access $(\alpha_M, a_M, 0, U_M)$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon > 0$, and a unitary black box V, then an ϵ-additive approximation $\tilde{u}$ to $u := \langle 0|V^\dagger M V|0\rangle$ can be returned, with probability at least $\frac{2}{3}$, using $O(\alpha_M/\epsilon)$ queries to U_M, $O(\alpha_M/\epsilon)$ queries to V, as well as $O((n + a_M)\alpha_M/\epsilon)$ additional 2-qubit gates.

Proof. We prove this theorem by constructing Algorithm 2, and now proceed to prove its correctness and complexity.

Correctness. Define $r := |\langle +|\langle 0|V^\dagger M' V|+\rangle|0\rangle|$. Then $\tilde{r}$ is an $(\epsilon/2\alpha_M)$-additive approximation to r with probability at least $\frac{2}{3}$. By Lemma 3, using α_M as an upper bound on $\|M\|$, we have $u = \alpha_M(2r - 1)$. Therefore, $\tilde{u}$ is an ϵ-additive approximation to u with probability at least $\frac{2}{3}$.

Complexity: As the algorithm makes only a single call to amplitude estimation, the complexities follow directly from Lemmas 2 and 3. ■

Algorithm 2 concludes our review of algorithms for estimating expectation values of block-encoded matrices.

III. APPROACH

In this section, we describe our formulation of SLEP in the block-encoding setting, our methods for establishing query-complexity lower bounds for this computational problem, and the techniques we use in constructing a quantum algorithm that saturates this bound. The detailed derivation of our results is presented in Sec. IV.

A. Query-complexity lower bound for B-EVHM

We begin by presenting our computational problem statements, starting with B-SLEP, which is a formulation of SLEP in the block-encoding setting. We then discuss two expectation-value-estimation problems. We close by describing the reductions we use to derive our lower bounds for B-SLEP.

In contrast to QLSP (Problem 1), the output of SLEP is a number, not a quantum state. We formulate SLEP in the block-encoding setting.

Problem 5 (Block-access system of linear equations problem (B-SLEP)). Given n qubits, a $\kappa \geqslant 1$, block access $(\alpha_A, a_A, 0, U_A)$ to a $2^n \times 2^n$ invertible Hermitian matrix A such that $\|A^{-1}\|\alpha_A \leqslant \kappa$, block access $(\alpha_M, a_M, 0, U_M)$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon \in [\alpha_M/2^n, \alpha_M]$, and an n-qubit unitary black box $U_{\boldsymbol{b}}$, return with probability at least $\frac{2}{3}$ an ϵ-additive approximation to $\boldsymbol{x}^\dagger M \boldsymbol{x}$, where $\boldsymbol{x} := A^{-1}\boldsymbol{b}$ and $\boldsymbol{b}$ is the 2^n-dimensional complex vector with entries $b_i = \langle i|U_{\boldsymbol{b}}|0^n\rangle$.

We require that $\epsilon \leqslant \alpha_M$ as otherwise the problem is trivial with 0 a valid answer. We restrict to instances with $\epsilon \geqslant \alpha_M/2^n$, as our query-complexity lower bound only applies to these instances. Additionally, by Theorem 3, any quantum algorithm solving B-SLEP for instances with $\epsilon \leqslant \alpha_M/2^n$ must have complexity superpolynomial in n across these instances, under complexity-theoretic assumptions. As a result, the requirement $\epsilon \geqslant \alpha_M/2^n$ restricts only to instances for which a quantum algorithm could have complexity in poly(n). Note that α_A and α_M bound the norm of A and M, as blocks of U_A and U_M do not have norm greater than 1.

It follows from the Chernoff bound ([31], p. 154) that a constant number of repetitions of an algorithm solving Problem 5 is sufficient to boost the success probability to a constant $s \in [\frac{2}{3}, 1)$ using a number of repetitions that are logarithmic in $1/(1-s)$. Therefore, if the bound $\frac{2}{3}$ on the success probability is replaced by any constant in the interval $[\frac{2}{3}, 1)$ no changes in complexity occur. In any problem where the output is an ϵ-additive approximation, we require that the output of the algorithm in the failed attempts is a complex number but not necessarily an ϵ-additive approximation. Although A is assumed to be Hermitian in B-SLEP, our results also apply to non-Hermitian matrices, as they can be encoded into Hermitian matrices [6].

We aim to establish a lower bound on the number of queries to U_M that any quantum algorithm for B-SLEP must make. As the solution state $|\boldsymbol{x}\rangle$ can be generated using algorithms for QLSP without making any queries to U_M, we focus on the measurement step where queries to U_M are required. The measurement step is in fact formalized by the problem B-EVHM discussed in Sec. II E 3. This problem is equivalent to B-SLEP restricted to the case $A = \mathbb{1}$. Therefore, a lower bound for B-EVHM gives a lower bound for B-SLEP.

Our lower bound for B-EVHM is found by first proving a lower bound on the analog of B-EVHM in the sparse-access setting, which we state below.

Problem 6 (Sparse-access expectation value of a Hermitian matrix (S-EVHM)). Given number of qubits n, block access $(d_M, \beta_M, O_{M_{\text{val}}}, O_{M_{\text{loc}}})$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon \in [d_M\beta_M/2^n, d_M\beta_M]$, and an n-qubit unitary black box V, return with probability at least $\frac{2}{3}$ an ϵ-additive approximation to $\langle 0^n|V^\dagger M V|0^n\rangle$.

To establish our query-complexity lower bound on S-EVHM, we reduce the problem of calculating the approximate mean of a black-box function (Problem 3) to S-EVHM. This reduction allows us to use the known bounds for AM [30], given in Corollary 2, to derive bounds for S-EVHM. Furthermore, as S-EVHM is reducible to B-EVHM, and since B-SLEP is reducible to B-EVHM, we can extend these lower bounds to both B-EVHM and B-SLEP. The reduction of AM to S-EVHM relies on the observation that the expectation value of a Hermitian matrix M with respect to the uniform superposition state $|+^n\rangle := 1/\sqrt{N}\sum_{j\in[N]}|j\rangle$ equals the mean of the entries of M, i.e.,

$$\langle +^n|M|+^n\rangle = \frac{1}{N}\sum_{i,j\in[N]} M_{ij}. \tag{21}$$

Given oracle access to a function $f : [N] \to \{0, 1\}$, we construct sparse access to a matrix M that encodes the function f in its entries. The mean of f can then be calculated by one call to S-EVHM with $U = H^{\otimes n}$, which completes the reduction. A function f can be encoded in the entries of a Hermitian matrix in many different ways, with one way being encoding the function values along the diagonal entries of the matrix. Whereas such an encoding would be sufficient to derive a lower bound for S-EVHM with respect to ϵ and the max-norm β_M, we employ a different reduction to achieve a lower bound with respect to ϵ, d_M, and β_M. We encode a function on domain $[d_M 2^{n-1}]$ in the entries of a Hermitian matrix of sparsity d_M. We use this encoding to prove that the query complexity of S-EVHM is $\Omega(d_M\beta_M/\epsilon)$. Our query-complexity lower bound for B-EVHM is derived by a reduction leveraging known methods for constructing block encodings from sparse-access encodings given in Lemma 1. For B-EVHM, S-EVHM, and B-SLEP, the query-complexity lower bound yields $\Omega(1/\epsilon)$ dependence on precision, which rules out the possibility of any algorithm solving any of these problems in superlogarithmic time.

We note that a lower bound on B-EVHM could be established directly from the lower bound for AM. However, we opt to first prove a lower bound on S-EVHM and then derive our lower bound on B-EVHM. Our motivation for this choice is that the lower bound for S-EVHM is important beyond its application to deriving a lower bound for B-SLEP, as estimation of expectation value of a sparse Hermitian matrix is a standard subroutine in several algorithms [24,32,34–36].

B. Tightness of the lower bound

Our approach for proving the tightness of our query-complexity lower bounds for B-EVHM, B-SLEP, and S-EVHM is to give quantum algorithms that achieve these bounds. The algorithm for B-EVHM discussed in Sec. II E 3 saturates our bound for B-EVHM. We now describe how we design our quantum algorithms for B-SLEP and S-EVHM. The detailed construction of these algorithms, along with their complexities, is given in Sec. IV B and Appendix B.

We start by describing a reduction of B-SLEP to B-EVHM, and then describe our quantum algorithm for B-EVHM. This reduction employs a block encoding of A^{-1}, which is known to be implementable with polylog$(1/\epsilon)$ queries to U_A using techniques reviewed in Sec. II B 1. Let $U_{A^{-1}}$ be an $(\alpha, a, 0)$ block encoding of A^{-1}. Observe that

$$\langle 0^a|\langle \boldsymbol{b}|U_{A^{-1}}^{\dagger}(|0^a\rangle\langle 0^a| \otimes M)U_{A^{-1}}|0^a\rangle|\boldsymbol{b}\rangle = \frac{1}{\alpha^2}\langle \boldsymbol{b}|A^{-1\dagger}MA^{-1}|\boldsymbol{b}\rangle = \frac{1}{\alpha^2}\boldsymbol{x}^{\dagger}M\boldsymbol{x}. \quad (22)$$

A block encoding of $|0^a\rangle\langle 0^a| \otimes M$ is implementable by a simple circuit that queries the block encoding of M. The left-hand side of Eq. (22) is the expectation value of $|0^a\rangle\langle 0^a| \otimes M$ with respect to $U_{A^{-1}}|0^a\rangle|\boldsymbol{b}\rangle$, and therefore can be estimated by one call to an algorithm for B-EVHM. This completes the reduction of B-SLEP to B-EVHM.

To design our algorithm for solving S-EVHM, we use a known procedure for generating a block encoding of M using the sparse-access oracles for M (Lemma 1). This block encoding of M is used as an input to our algorithm for B-EVHM, which completes the design of our algorithm for S-EVHM. This concludes the discussion of our approach.

IV. RESULTS

In this section we prove the query complexity of B-SLEP is $\Theta(\alpha_M/\epsilon)$. In Sec. IV A we give a lower bound for B-SLEP. Next, in Sec. IV B we describe a quantum algorithm that saturates this bound.

A. Lower bound for B-SLEP

In this section, we derive query-complexity lower bounds for B-EVHM, S-EVHM, and B-SLEP. We first derive a lower bound on queries to U_M for S-EVHM, which utilizes existing query-complexity lower bound for AM [30]. Next, we use Lemma 1 to reduce S-EVHM to B-EVHM and prove a lower bound for B-EVHM. We extend this lower bound to B-SLEP by reducing B-EVHM to instances of B-SLEP with $A = \mathbb{1}$.

We now derive a lower bound for S-EVHM by a reduction of AM to S-EVHM. En route to this reduction, we define a scaled version of AM and obtain the dependence of its query complexity on the scaling factor and the accuracy.

Problem 7 (Scaled approximate mean (SAM)). Given $N \in \mathbb{Z}^+$, $\epsilon \in [\beta/2N, \beta)$, $\beta \in \mathbb{R}^+$ and access to an oracle Q_g encoding a function $g : [N] \mapsto [0, \beta]$, return an approximation $\tilde{\mu}_g \in [0, \beta]$ of the mean $\mu_g := (\sum_{j=0}^{N-1} g(j))/N$ such that $|\tilde{\mu}_g - \mu_g| < \epsilon$ with probability at least $\frac{2}{3}$.

Lemma 5. Given an algorithm $\mathcal{Q}$ that solves SAM for any valid input $(N', 1/\epsilon', \beta', O_g)$ by making K queries to O_g, an algorithm $\mathcal{R}$ can be constructed that solves AM for input $(N', \beta'/\epsilon', O_f)$ by making K queries to O_f.

Proof. Using queries to $\mathcal{Q}$, we design a algorithm $\mathcal{R}$ to solve the AM instance given by the input in the lemma statement. Using the input oracle O_f, construct a new oracle O_g encoding the function $g : [N] \mapsto [0, \beta'] : j \mapsto \beta' f(j)$. The algorithm $\mathcal{R}$ first obtains a number $\tilde{\mu}_g$ by running the algorithm $\mathcal{Q}$ with input $(N', 1/\epsilon', \beta', O_g)$, and then returns $\tilde{\mu}_f = \tilde{\mu}_g/\beta'$.

The number $\tilde{\mu}_g$ satisfies $|\tilde{\mu}_g - \mu_g| < \epsilon'$ with probability at least $\frac{2}{3}$, where μ_g is the mean of g as defined in Problem 7. Therefore, with probability at least $\frac{2}{3}$, $\tilde{\mu}_f$ satisfies $|\tilde{\mu}_f - \mu_f| < \epsilon'/\beta'$ as required. Furthermore, each call to O_g can be implemented with one call to O_f, therefore, $\mathcal{R}$ makes K queries to O_f. ■

We prove the complexity of AM implies a lower bound for SAM.

Lemma 6. The query complexity of SAM is $\Omega(\beta/\epsilon)$.

Proof. From Corollary 2, there exist constants $\eta_0, C_0 \in \mathbb{R}^+$ such that any quantum algorithm solving AM on input $(N, 1/\epsilon, O_f)$, satisfying $N, 1/\epsilon > \eta_0$, makes at least C_0/ϵ queries to O_f. Our proof is by contradiction, so assume SAM has query complexity not in $\Omega(\beta/\epsilon)$. Then, there exists a quantum algorithm $\mathcal{Q}$ that solves SAM for some input $(N', 1/\epsilon', \beta', O_g)$, with $N', \beta', 1/\epsilon' > \max(\eta_0, 1)$, using fewer than $\lfloor C_0\beta'/\epsilon' \rfloor$ queries to O_g. By Lemma 5, there exists an algorithm $\mathcal{R}$ solving AM on input $(N', \beta'/\epsilon', O_g)$ which makes fewer than $\lfloor C_0\beta'/\epsilon' \rfloor = \lfloor C_0/\epsilon' \rfloor$ queries to O_g. As $N', \beta'/\epsilon', > \eta_0$, we have a contradiction to the first statement of the proof. ■

To prove the lower bound on the query complexity of S-EVHM, we make use of an encoding of a function into a sparse Hermitian matrix. We first define a function ind that maps each pair of a row and a column index of a $2^n \times 2^n$ Hermitian matrix M to an integer in $[2^{2n}]$.

Definition 5. For $n \in \mathbb{Z}^+$, we define the index function

$$\text{ind} : [2^n] \times [2^n] \to [2^{2n}]$$
$$(i, j) \mapsto 2^n\big((j - i) \bmod 2^n\big) + i. \quad (23)$$

Here mod$2^n : \mathbb{Z} \to [2^n]$ is the modulo operation. For example, for $n = 2$, the $2^n \times 2^n$ matrix B defined by $B_{ij} = \text{ind}(i, j)$ takes the form

$$B = \begin{pmatrix} 0 & 4 & 8 & 12 \\ 13 & 1 & 5 & 9 \\ 10 & 14 & 2 & 6 \\ 7 & 11 & 15 & 3 \end{pmatrix}. \quad (24)$$

We prove another lemma before explaining our encoding.

Lemma 7. For $n \in \mathbb{Z}^+$

(1) ind is invertible.

(2) for $i, j \in [2^n]$ such that $i \neq j$,

$$\text{ind}(i, j) + \text{ind}(j, i) \geqslant 2^{2n}. \quad (25)$$

Proof. We prove the two statements separately.

(1) The function ind is invertible if $\text{ind}(i, j) = \text{ind}(k, l)$ implies $i = k$, $j = l$, which is equivalent to

$$2^n[(j - i) \bmod 2^n - (l - k) \bmod 2^n] = k - i. \quad (26)$$

Comparing the left- and right-hand sides of this equation together with the constraint $|k - i| \leqslant 2^n$ readily yields $i = k$ and $j = l$.

(2) By definition of the ind function,

$$\text{ind}(i, j) + \text{ind}(j, i) = 2^n[(j - i) \bmod 2^n + (i - j) \bmod 2^n\big] + i + j. \quad (27)$$

The result follows readily from the observation $[(j - i) \bmod 2^n + (i - j) \bmod 2^n] = 2^n$ for $i \neq j$. ■

We are now ready to state the encoding of a function into a matrix that we use for our reduction.

Definition 6. For $n \in \mathbb{Z}^+$, $d_M \in [2^n]$, $\beta \in \mathbb{R}^+$, and a function $g : [d_M 2^{n-1}] \to [0, \beta]$, the (n, d_M)-matrix encoding of g

is the $2^n \times 2^n$ matrix M with entries

$$M_{ij} := \begin{cases} g(\mathrm{ind}(i,i)) & \text{if} \quad \mathrm{ind}(i,i) \in [d_M 2^{n-1}] \text{ and } \ i = j, \\ g(\mathrm{ind}(i,j))/2 & \text{if} \quad \mathrm{ind}(i,j) \in [d_M 2^{n-1}] \text{ and } \ i \neq j, \\ g(\mathrm{ind}(j,i))/2 & \text{if} \quad \mathrm{ind}(j,i) \in [d_M 2^{n-1}] \text{ and } \ i \neq j, \\ 0 & \text{if} \quad \mathrm{ind}(i,j), \mathrm{ind}(j,i) \notin [d_M 2^{n-1}] \text{ and } \ i \neq j. \end{cases} \tag{28}$$

Lemma 7 implies that for any $i \neq j$, both $\mathrm{ind}(i,j)$ and $\mathrm{ind}(j,i)$ cannot simultaneously belong to $[d_M 2^{n-1}]$, so each entry of M is unique. We now derive relevant properties of this matrix encoding.

Lemma 8. For $n \in \mathbb{Z}^+$, $d_M \in [2^n]$, $\beta \in \mathbb{R}^+$, and a function $g : [d_M 2^{n-1}] \to [0, \beta]$, let $2^n \times 2^n$ d_M-sparse matrix M be the (n, d_M)-matrix encoding of g. Then

(1) M is a Hermitian matrix,

(2) M is d_M sparse, and

(3) the entries of M satisfy

$$\sum_{i,j\in[2^n]} M_{ij} = \sum_{k\in[d_M 2^{n-1}]} g(k). \tag{29}$$

Proof. We provide a separate proof for each of the three statements.

(1) First we prove that M is Hermitian. As all entries of M are real, we only need to prove that $M_{ij} = M_{ji}$ for $i \neq j$. Suppose i, j are such that $\mathrm{ind}(i,j) \in [d_M 2^{n-1}]$. Then $\mathrm{ind}(j,i) \notin [d_M 2^{n-1}]$ by Lemma 7. Therefore, by Eq. (28), $M_{ij} = M_{ji} = g(\mathrm{ind}(i,j))/2$. A similar argument yields $M_{ij} = M_{ji}$ if $\mathrm{ind}(i,j) \notin [d_M 2^{n-1}]$.

(2) To prove that M is d_M sparse, we count, for each row i, the number of column indices $\{j\}$ for which M_{ij} is nonzero. For an entry M_{ij} to be nonzero, either $\mathrm{ind}(i,j)$ or $\mathrm{ind}(j,i)$ must belong to $[d_M 2^{n-1}]$. We first count the number of $\{j\}$ such that $\mathrm{ind}(i,j) \in [d_M 2^{n-1}]$. Note that for $(j - i) \mod 2^n > d_M/2$, we have $\mathrm{ind}(i,j) > d_M 2^{n-1}$, therefore, there are at most $\lceil d_M/2 \rceil$ values of j satisfying $\mathrm{ind}(i,j) \in [d_M 2^{n-1}]$ for any i. Similar argument can be made to show that there are at most $\lceil d_M/2 \rceil$ values of j for which $\mathrm{ind}(j,i) \in [d_M 2^{n-1}]$. Considering that $j = i$ features in both these lists, we conclude that any row i has at most d_M nonzero entries.

(3) This property is a consequence of the invertibility of ind and the Hermiticity of M. Every $k \in [d_M 2^{n-1}]$ in the domain of g is mapped to a pair of a row and a column index (i, j) by the inverse of the ind function. Furthermore, if $i \neq j$, then $M_{ij} = g(k)/2$ so that $M_{ij} + M_{ji} = g(k)$ by Hermiticity of M. ■

The (n, d_M)-matrix encoding M of g can be constructed with a single query to the function oracle, as given by the following lemma.

Lemma 9. Given n qubits, a sparsity parameter $d_M \in [2^n]$, a number $\beta_M \in \mathbb{R}^+$, and an oracle O_g encoding a function $g : [d_M 2^{n-1}] \to [0, \beta]$, sparse access $(d_M, \beta_M, O_{M_{\mathrm{val}}}, O_{M_{\mathrm{loc}}})$ to the (n, d_M)-matrix encoding of g can be constructed with $O_{M_{\mathrm{val}}}$ making a single query to O_g and $O_{M_{\mathrm{loc}}}$ making zero queries to O_g.

Proof. The oracle for M_{val} can be constructed using the explicit formula Eq. (28) for the entries of M. This construction requires first computing $\mathrm{ind}(i,j)$, then computing $g(\mathrm{ind}(i,j))$ by making one query to O_g, and finally uncomputing $\mathrm{ind}(i,j)$. The function $\mathrm{ind}(i,j)$ can be computed using the explicit formula Eq. (23) and requires no queries to O_g.

To construct an oracle for M_{loc}, it suffices to provide an explicit formula for computing M_{loc}. Recall that

$$\begin{aligned} M_{\mathrm{loc}} &: [2^n] \times [d_M] \to [2^n] \\ &: (i, l) \mapsto M_{\mathrm{loc}}(i, l), \end{aligned} \tag{30}$$

where $M_{\mathrm{loc}}(i, l)$ is the lth nonzero entry in the ith row of M. From the proof of the second statement of Lemma 8, we deduce that M_{loc} is given by $M_{\mathrm{loc}}(j, l) = i + l - \lfloor d_M/2 \rfloor \mod 2^n$. ■

The function to matrix encoding in Definition 6 ensures that the mean of a function g can be inferred from the expectation value of the (n, d_M)-matrix encoding of g with respect to the equal superposition state, as stated in the following lemma.

Lemma 10. Let $n \in \mathbb{Z}^+$, $d_M \in [2^n]$, $\beta \in \mathbb{R}^+$, $g : [d_M 2^{n-1}] \to [0, \beta]$ and let M be the (n, d_M)-matrix encoding of g. Then the mean μ_g of g satisfies

$$\mu_g = \frac{2}{d_M} \langle +^n | M | +^n \rangle, \tag{31}$$

where $|+^n\rangle = H^n |0^n\rangle$.

Proof. By Lemma 8, $\sum_{i,j\in[2^n]} M_{ij} = \sum_{i\in[d_M 2^{n-1}]} g(i)$. Therefore,

$$\begin{aligned} \langle +^n | M | +^n \rangle &= \frac{1}{2^n} \sum_{i,j\in[2^n]} M_{ij} \\ &= \frac{1}{2^n} \sum_{i\in[d_M 2^n]} g(i) = \frac{d_M 2^{n-1}}{2^n} \mu_g \qquad (32) \\ &= \frac{d_M}{2} \mu_g, \qquad (33) \end{aligned}$$

where the second equality follows from Lemma 8. ■

We now give our reduction from SAM to S-EVHM.

Lemma 11. Given an algorithm $\mathcal{Q}$ that solves the problem S-EVHM on any instance

$$(n, 1/\epsilon, (d_M, \beta_M, O_{M_{\mathrm{val}}}, O_{M_{\mathrm{loc}}}), H^{\otimes n}) \tag{34}$$

by making K queries to the sparse-access oracles encoding M, an algorithm $\mathcal{R}$ can be constructed that solves the SAM instance with input

$$(d_M 2^{n-1}, d_M/2\epsilon, \beta_M, O_g), \tag{35}$$

by making K queries to O_g.

Proof. We show how to construct an algorithm $\mathcal{R}$ as described in the Lemma. Let M be the (n, d_M)-matrix encoding of g, which is given by Definition 6 and has sparse-access oracles given in Lemma 9. Let $\mathcal{R}$ be the algorithm that queries $\mathcal{Q}$ with input given by Eq. (34) to compute $\tilde{u}$ and returns $2\tilde{u}/d_M$. By Lemma 10, $\langle +^n | M | +^n \rangle = d_M \mu_g / 2$. Then $|\tilde{u} - d_M \mu_g / 2| < \epsilon$ with probability at least $\frac{2}{3}$, and so $|2\tilde{u}/d_M - \mu_g| < 2\epsilon/d_M$

with probability $\frac{2}{3}$. Therefore, $\mathcal{R}$ solves the SAM instance defined by the inputs in Eq. (35). Also, $\mathcal{R}$ makes K queries to O_g as $\mathcal{Q}$ makes K queries to oracles encoding M, and each query to M requires only one query to O_g by Lemma 9. ■

Using this reduction, we derive the following query-complexity lower bound.

Lemma 12. Given n qubits, sparse access $(d_M, \beta_M, O_{M_{\text{val}}}, O_{M_{\text{loc}}})$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon \in (d_M\beta/2^n, d_M\beta)$, and an n-qubit unitary black box V, any quantum algorithm that returns with probability at least $\frac{2}{3}$ an ϵ-additive approximation of $\langle 0|V^\dagger MV|0\rangle$ for $V = H^{\otimes n}$ makes $\Omega(d_M\beta/\epsilon)$ queries U_M.

Proof. From Lemma 6, there exist constants $C_0, \eta_0 \in \mathbb{R}^+$ such that any quantum algorithm solving the SAM problem with input $(N, 1/\epsilon, \beta, O_g)$, where $N, 1/\epsilon, \beta > \eta_0$, makes at least $\lfloor C_0\beta/\epsilon\rfloor$ queries to O_g. We proceed by contradiction. Assume Lemma 12 is false, then there exists a quantum algorithm $\mathcal{Q}$ which for some input

$$(n', (d'_M, \beta'_M, O_{M_{\text{val}}}, O_{M_{\text{loc}}}), 1/\epsilon') \tag{36}$$

with $n', d'_M, 1/\epsilon', \beta'_M > 2\eta_0$, returns an ϵ-additive approximation of $\langle 0|V^\dagger MV|0\rangle$ using a number of queries to oracles encoding M fewer than $\lfloor C_0 d'_M\beta'_M/2\epsilon'\rfloor$. Let $N'' := d'_M 2^{n'-1}$, $1/\epsilon'' := d'_M/2\epsilon_0$, and $\beta'' := \beta'_M$. Then by Lemma 11 there exists an algorithm $\mathcal{R}$ solving SAM on input

$$(N'', 1/\epsilon'', \beta'', O_g), \tag{37}$$

which makes fewer than $\lfloor C_0 d'_M\beta'_M/2\epsilon'\rfloor = \lfloor C_0\beta''/\epsilon''\rfloor$ queries to O_g. Using $n', d'_M, 1/\epsilon', \beta'_M > 2\eta_0$ we have $N'', 1/\epsilon'', \beta'' > \eta_0$. Therefore, the existence of $\mathcal{R}$ contradicts the first statement of the proof. ■

Lemma 12 addresses S-EVHM under the restriction $V = H^{\otimes n}$. This lemma implies the following lower bound for S-EVHM.

Corollary 3. Any algorithm that solves S-EVHM makes $\Omega(d_M\beta_M/\epsilon)$ queries to sparse-access oracles for M.

We next use Corollary 3 to prove a similar query-complexity lower bound for B-EVHM.

Theorem 6. Any algorithm that solves B-EVHM makes $\Omega(\alpha_M/\epsilon)$ queries to the block encoding of M.

Proof. From Theorem 12, there exist constants $\eta_0, C_0 \in \mathbb{Z}^+$ such that any quantum algorithm solving the problem S-EVHM on input $(n, (d_M, \beta_M, O_{M_{\text{val}}}, O_{M_{\text{loc}}}), 1/\epsilon, V)$, satisfying $n, d_M, \beta_M, 1/\epsilon > \eta_0$ makes at least $\lfloor C_0 d_M\beta_M/\epsilon\rfloor$ queries to M. Our proof is by contradiction. Assume that a quantum algorithm $\mathcal{Q}$ solves B-EVHM with queries to U_M not in $\Omega(\alpha_M/\epsilon)$. Then there exists an input $(n', (\alpha'_M, a'_M, 0, U_M), 2/\epsilon', V)$, with $n', \alpha'_M, a'_M, 1/\epsilon' > (\eta_0 + 1)^2$ and $a'_M \geqslant 2$, for which $\mathcal{Q}$ solves B-EVHM using fewer than $\lfloor C_0\alpha'_M/\epsilon'\rfloor$ queries to U_M. For

$$d'_M = \eta_0 + 1 \text{ and } \beta'_M = \alpha'_M/(\eta_0 + 1), \tag{38}$$

and following Lemma 1, block access $(d'_M\beta'_M, a'_M, \epsilon'/2, U_M)$ can be constructed such that U_M makes a single query to sparse-access oracles for M. Applying $\mathcal{Q}$ on the instance defined by input $(n', (d'_M\beta'_M, \alpha'_M, 0, U_M), 2/\epsilon', V)$ gives a quantum algorithm that solves S-EVHM with input $(n', (d'_M, \beta'_M, O_{M_{\text{val}}}, O_{M_{\text{loc}}}), 1/\epsilon', V)$ by making fewer than $\lfloor C_0 d'_M\beta'_M/\epsilon'\rfloor = \lfloor C_0\alpha_M/\epsilon_0\rfloor$ queries to M. As $n', d'_M, \beta'_M, 1/\epsilon' > \eta_0$ in the input to this instance of S-EVHM, we have a contradiction to the first statement of the proof. ■

Theorem 6 establishes a query-complexity lower bound for the problem B-EVHM, in which the matrix M is given via block encoding, and the state is generated by a unitary black box V. In Appendix A, we show that the lower bound in Theorem 6 also applies for the problem of estimating the expectation value even if V is any fixed unitary operator and not a black box. In particular for the restriction $V = \mathbb{1}_n$, the output of this problem is an estimate of $\langle 0^n|M|0^n\rangle$. Interestingly, the restriction of S-EVHM to $V = \mathbb{1}_n$ can be solved by making just one query to the sparse oracles for M.

As B-EVHM is a restriction of B-SLEP to $A = \mathbb{1}_n$, the lower bound in Theorem 6 immediately yields a lower bound for B-SLEP.

Theorem 7. Any quantum algorithm that solves B-SLEP must make $\Omega(\alpha_M/\epsilon)$ queries to U_M.

This theorem concludes our derivation of lower bound for B-SLEP.

B. Tightness of bound

In this section, we construct quantum algorithms for solving B-SLEP that achieves the query-complexity lower bounds derived in the previous section. Our algorithm for B-SLEP follows the approach outlined in Sec. III B. We use the algorithm for B-EVHM discussed in Sec. II E 3 as a subroutine to design an algorithm for B-SLEP in Sec. IV B. We first show how to construct a block encoding of A^{-1} using techniques discussed in Sec. II C 1. Our algorithm for B-SLEP makes one query to our algorithm for B-EVHM (Algorithm 2) and queries the block encoding of A^{-1} as an input to this single query.

First we combine Theorem 1 and Corollary 1 to show how block access to A^{-1} can be constructed using block access to A. A more general result in this direction is given in Ref. [23]. We provide a different derivation in our setting and report complexities for queries to block encoding of A and 2-qubit gates separately.

Corollary 4. Given n qubits, an accuracy $\epsilon \in (0, 2]$, a $\kappa \geqslant 1$, and block access $(\alpha_A, a_A, 0, U_A)$ to an invertible $2^n \times 2^n$ Hermitian matrix A such that $\|A^{-1}\|\alpha_A \leqslant \kappa$, then block access $(8\kappa/3, a_A + 2, \epsilon, U_{A^{-1}})$ to A^{-1} can be constructed such that $U_{A^{-1}}$ makes $O(\kappa\log(\kappa/\epsilon))$ queries to U_A and $O(\kappa a_A\log(\kappa/\epsilon))$ additional 2-qubit gates.

Proof. Invoking Corollary 1 with the theorem's δ parameter equal to $1/\kappa$ and the theorem's ϵ parameter equal to $3\epsilon/(8\kappa)$, there exists a polynomial $\mathcal{P}$ with degree in $O(\kappa\log(\kappa/\epsilon))$ such that for all

$$x \in [-1, 1] \setminus [-1/\kappa, 1/\kappa], \tag{39}$$

$|\mathcal{P}(x)/2| \leqslant 1/2$ and

$$\left|\frac{\mathcal{P}(x)}{2} - \frac{3}{8\kappa}\frac{1}{x}\right| < \frac{3\epsilon}{16\kappa}. \tag{40}$$

As the spectrum of A/α_A lies in $[1, -1] \setminus [-1/\kappa, 1/\kappa]$, each eigenvector $|\lambda\rangle$ of A satisfies

$$\left|\langle\lambda|\left(\frac{3}{8\kappa}A^{-1} - \frac{\mathcal{P}(A/\alpha_A)}{2}\right)|\lambda\rangle\right| \leqslant \frac{3\epsilon}{16\kappa} \tag{41}$$

and, consequently,

$$\left\| \frac{3}{8\kappa}A^{-1} - \frac{\mathcal{P}(A/\alpha_A)}{2} \right\| \leqslant \frac{3\epsilon}{16\kappa}. \tag{42}$$

Next, Theorem 1 with $\sigma = 3\epsilon/16\kappa$ guarantees that a $(1, a_A + 2, 3\epsilon/16\kappa)$ block encoding of $\mathcal{P}(A/\alpha_A)/2$ can be constructed using $O(\kappa \log(\kappa/\epsilon))$ queries to U_A and $O(\kappa a_A \log(\kappa/\epsilon))$ additional 2-qubit gates. From Eq. (42), this block encoding of $\mathcal{P}(A/\alpha_A)/2$ also block encodes A^{-1} with the prefactor $8\kappa/3$ and error bounded by

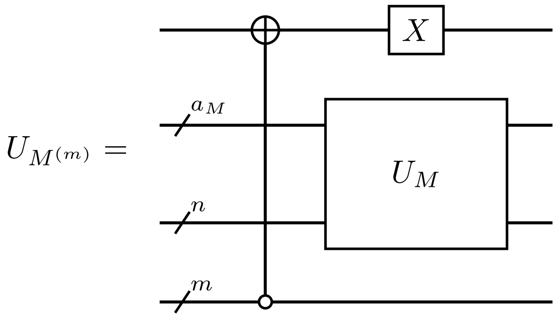
$$(8\kappa/3)(3\epsilon/16\kappa + 3\epsilon/16\kappa) = \epsilon. \tag{43}$$

Therefore, the block encoding of $\mathcal{P}(A/\alpha_A)/2$ given by Theorem 1 is the required block access $(8\kappa/3, a_A + 2, \epsilon, U_{A^{-1}})$ to A^{-1}. ■

We next prove that block access to M can be used to construct block access to $M^{(m)} := M \otimes |0^m\rangle\langle 0^m|$ for any given $m \in \mathbb{Z}^+$.

Lemma 13. Given n-qubits, block access $(\alpha_M, a_M, 0, U_M)$ to a $2^n \times 2^n$ Hermitian matrix M, and given an $m \in \mathbb{Z}^+$, then block access $(\alpha_M, a_M + 1, 0, U_{M^{(m)}})$ to $M^{(m)}$ can be constructed such that $U_{M^{(m)}}$ queries U_M once and employs $O(m^2)$ additional 2-qubit gates.

Proof. We show that the desired block encoding of $M^{(m)}$ is given by the $(1 + a_M + n + m)$-qubit circuit



The controlled-NOT (CNOT) gate in this circuit has m control qubits, and only flips the target qubit if the control qubits are in the state $|0^m\rangle$. For any computational basis states $|0^{a_M+1}, s_1, 0^m\rangle$ and $|0^{a_M+1}, s_2, 0^m\rangle$, where s_1 and s_2 are n-bit strings,

$$\langle 0^{a_M+1}, s_1, 0^m|U_{M^{(m)}}|0^{a_M+1}, s_2, 0^m\rangle = M_{s_1 s_2}/\alpha_M \tag{44}$$

is true. Furthermore, for any states $|0^{a_M+1}, s_1, r_1\rangle$ and $|0^{a_M+1}, s_2, r_2\rangle$, where r_1 and r_2 are m-bit strings such that at least one of r_1 and r_2 is not equal to 0^m,

$$\langle 0^{a_M+1}, s_1, r_1|U_{M^{(m)}}|0^{a_M+1}, s_2, r_2\rangle = 0. \tag{45}$$

Equations (44) and (45) together imply

$$\langle 0^{a_M+1}|U_{M^{(m)}}|0^{a_M+1}\rangle = (M \otimes |0^m\rangle\langle 0^m|)/\alpha_M, \tag{46}$$

as required. The required number of 2-qubit gates scales as $O(m^2)$, which follows from the fact that the 0^m-CNOT gate can be implemented using $O(m)$ 2-qubit gates and the Toffoli gate [31]. ■

Theorem 8. A quantum algorithm can be constructed that takes n qubits, a $\kappa \geqslant 1$, block access $(\alpha_A, a_A, 0, U_A)$ to a $2^n \times 2^n$ invertible Hermitian matrix A such that $\|A^{-1}\|\alpha_A \leqslant \kappa$, block access $(\alpha_M, a_M, 0, U_M)$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon \in [\alpha_M/2^n, \alpha_M]$, and an n-qubit unitary black box $U_{\boldsymbol{b}}$, and returns with probability at least $\frac{2}{3}$ an ϵ-additive approximation to $\boldsymbol{x}^\dagger M\boldsymbol{x}$, where $\boldsymbol{x} := A^{-1}\boldsymbol{b}$ and $\boldsymbol{b}$ is the 2^n-dimensional vector with entries $b_i = \langle i|U_{\boldsymbol{b}}|0\rangle$, by making $O(\alpha_M\kappa^2/\epsilon)$ queries to U_M, $O(\alpha_M\kappa^3 \log(\alpha_M\kappa^2/\epsilon)/\epsilon)$ queries to U_A, $O(\alpha_M\kappa^2/\epsilon)$ queries to $U_{\boldsymbol{b}}$, and

$$O\left\{\frac{\alpha_M\kappa^2}{\epsilon}\left[n + a_M + a_A\kappa \log\left(\frac{\alpha_M\kappa^2}{\epsilon}\right) + a_A\right]\right\}, \tag{47}$$

additional 2-qubit gates.

Proof. We describe an algorithm (Algorithm 3) for B-SLEP, and prove the correctness of this algorithm. This algorithm makes reference to subroutines described in other work. We end the proof by analyzing the algorithm's complexity. The algorithm is as follows.

Correctness. Let

$$\begin{aligned} r :=\ & \langle 0^{n+a_A+2}|(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}}^\dagger)(U_{A^{-1}}^\dagger)(|0^{a_A+2}\rangle\langle 0^{a_A+2}| \otimes M) \\ & \times (U_{A^{-1}})(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}})|0^{n+a_A+2}\rangle. \end{aligned} \tag{48}$$

The output of B-EVHM (Algorithm 2) satisfies

$$|r - \tilde{r}| \leqslant (\epsilon/2)(8\kappa/3)^{-2}, \tag{49}$$

with probability at least $\frac{2}{3}$. Let $u := \langle \boldsymbol{x}|M|\boldsymbol{x}\rangle$, where $|\boldsymbol{x}\rangle$ is the solution state to $A\boldsymbol{x} = \boldsymbol{b}$, which can be expressed as

$$u = \langle \boldsymbol{b}|{A^{-1}}^\dagger MA^{-1}|\boldsymbol{b}\rangle = \langle 0^n|U_{\boldsymbol{b}}^\dagger {A^{-1}}^\dagger MA^{-1}U_{\boldsymbol{b}}|0^n\rangle. \tag{50}$$

Define $\tilde{E} := (8\kappa/3)\langle 0^{a_A+2}|U_{A^{-1}}|0^{a_A+2}\rangle$. Then, by the definition of block encoding (Definition 2), we have

$$\|A^{-1} - \tilde{E}\| \leqslant \epsilon/8\gamma\kappa. \tag{51}$$

Furthermore, this bound gives $\tilde{E}|\boldsymbol{b}\rangle - A^{-1}|\boldsymbol{b}\rangle \leqslant \epsilon/8\gamma\kappa$. Analyzing Eq. (48) yields $(8\kappa/3)^2 r = \langle \boldsymbol{b}|\tilde{E}^\dagger M\tilde{E}|\boldsymbol{b}\rangle$. Therefore,

$$\begin{aligned} |u - (8\kappa/3)^2 r| &= |\langle \boldsymbol{b}|{A^{-1}}^\dagger MA^{-1}|\boldsymbol{b}\rangle - \langle \boldsymbol{b}|\tilde{E}^\dagger M\tilde{E}|\boldsymbol{b}\rangle| \\ &\leqslant \|M\|(\|\tilde{E}|\boldsymbol{b}\rangle\|^2 - \|A^{-1}|\boldsymbol{b}\rangle\|^2|) \\ &\leqslant \|M\|(|\|\tilde{E}|\boldsymbol{b}\rangle\| - \|A^{-1}|\boldsymbol{b}\rangle\||) \\ &\times (\|\tilde{E}|\boldsymbol{b}\rangle\| + \|A^{-1}|\boldsymbol{b}\rangle\|) \\ &\leqslant \alpha_M\frac{\epsilon}{8\gamma\kappa}\left[\left(\frac{\epsilon}{8\gamma\kappa} + \kappa\right) + \kappa\right] \\ &\leqslant \frac{\alpha_M}{\gamma^2}\frac{\epsilon^2}{64\kappa^2} + \frac{\alpha_M}{\gamma}\frac{\epsilon}{4} \leqslant \epsilon/2, \end{aligned} \tag{52}$$

where the inequalities $\alpha_M/\gamma, \alpha_M/\gamma^2 \leqslant 1$ are used in the last step. By the triangle inequality,

$$\begin{aligned} |\tilde{u} - u| &\leqslant |\tilde{u} - (8\kappa/3)^2 r| + |u - (8\kappa/3)^2 r| \\ &= |(8\kappa/3)^2\tilde{r} - (8\kappa/3)^2 r| + |u - (8\kappa/3)^2 r| \\ &\leqslant \epsilon/2 + \epsilon/2 = \epsilon, \end{aligned} \tag{53}$$

with probability at least $\frac{2}{3}$.

Complexity. We consider the instances with $\alpha_M \geqslant 1$, which are sufficient to determine big-O scaling for queries and 2-qubit gates used by our algorithm. For these instances, Algorithm 3. Algorithm for block-access system of linear equations problem.

```
Input: $(n, \kappa, (\alpha_A, a_A, 0, U_A), (\alpha_M, a_M, 0, U_M), \epsilon, U_{\boldsymbol{b}})$
Output: With probability at least $\frac{2}{3}$ an $\epsilon$-additive approximation to $\boldsymbol{x}^\dagger M \boldsymbol{x}$, where $\boldsymbol{x} := A^{-1}\boldsymbol{b}$ and $\boldsymbol{b}$ is the vector with entries $b_i = \langle i|U_{\boldsymbol{b}}|0\rangle$
Procedure:
 1: $\gamma = \begin{cases} \alpha_M \text{ if } \alpha_M \geqslant 1 \\ \sqrt{\alpha_M} \text{ if } \alpha_M < 1. \end{cases}$
 2: Construct block access $(8\kappa/3, a_A + 2, \epsilon/8\gamma\kappa, U_{A^{-1}})$ to $A^{-1}$.            ▷ Use Corollary 4
 3: Construct block access $(\alpha_M, a_M + 1, 0, U_{M^{(a_A+2)}})$ to $M^{(a_A+2)} := |0^{a_A+2}\rangle\langle 0^{a_A+2}| \otimes M$.   ▷ Use Lemma 13
 4: Compute $\tilde{r}$ by executing the algorithm for B-EVHM with input                          ▷ Use Algorithm 2
        $(n + a_A + 2,\ (\alpha_M, a_M + 1, 0, U_{M^{(a_A+2)}}),\ (\epsilon/2)(8\kappa/3)^{-2},\ (U_{A^{-1}})(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}}))$.
 5: return $\tilde{u} = (8\kappa/3)^2\tilde{r}$.
```

$\gamma = \alpha_M$ by the first line of Algorithm 3. Using the complexities from Theorem 5, the single call made to Algorithm 3 makes

$$O(\alpha_M 2(8\kappa/3)^2/\epsilon) = O(\alpha_M \kappa^2/\epsilon), \tag{54}$$

queries to $U_{A^{-1}}(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}})$ and $U_{M^{(a_A+2)}}$. Therefore, using the complexity of $U_{A^{-1}}$ given by Corollary 4, the algorithm makes $O(\alpha_M \kappa^2/\epsilon)$ queries to each of $U_{\boldsymbol{b}}$, U_M, and $U_{A^{-1}}$. Each query to $U_{A^{-1}}$ comprises

$$O(\kappa \log(8\alpha_M \kappa^2/\epsilon)) = O(\kappa \log(\alpha_M \kappa^2/\epsilon)), \tag{55}$$

queries to U_A. Thus, our algorithm makes $O(\alpha_M \kappa^3 \log(\alpha_M \kappa^2/\epsilon)/\epsilon)$ queries to U_A.

Now we count the additional 2-qubit gates used by our algorithm. We separately count 2-qubit gates used directly by Algorithm 2, 2-qubit gates used by queries to $U_{A^{-1}}(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}})$), and 2-qubit gates used by queries to $U_{M^{(a_A+2)}}$. We then combine these three contributions to yield the total number of 2-qubit gates used. First, the number of 2-qubit gates used directly by Algorithm 2 is

$$O(\alpha_M \kappa^2(n + a_A + 2 + a_M)/\epsilon) = O(\alpha_M \kappa^2(n + a_A + a_M)/\epsilon). \tag{56}$$

Next, $U_{A^{-1}}(\mathbb{1}_{a_A+2} \otimes U_{\boldsymbol{b}})$ is queried $O(\alpha_M \kappa^2/\epsilon)$ times by B-EVHM. By Corollary 4, $U_{A^{-1}}$ uses

$$O(\kappa a_A \log(8\alpha_M \kappa^2/\epsilon)) \in O(\kappa a_A \log(\alpha_M \kappa^2/\epsilon)) \tag{57}$$

additional 2-qubit gates. Consequently,

$$O(\alpha_M \kappa^3 a_A \log(\alpha_M \kappa^2/\epsilon)/\epsilon) \tag{58}$$

2-qubit gates are used throughout all queries to $U_{A^{-1}}(\mathbb{1}_{(a_A+2)} \otimes U_{\boldsymbol{b}})$. Finally, we count the number of 2-qubit gates used in queries to $U_{M^{(a_A+2)}}$. By Lemma 13, each query to the oracle $U_{M^{(a_A+2)}}$ requires $O(a_A)$ gates, and this oracle is queried $O(\kappa^2 \alpha_M/\epsilon)$ times by B-EVHM, which yields a total of

$$O(a_A \alpha_M \kappa^2/\epsilon) \tag{59}$$

additional 2-qubit gates used throughout calls to $U_{M^{(a_A+2)}}$. Combining these three contributions (56)–(59) yields the total number of additional 2-qubit gates given by Eq. (47). ■

In comparison with algorithms for QLSP [8], which have linear dependence on κ, our algorithm has cubic dependence on κ. This additional dependence on κ comes from the added challenge of solving B-SLEP over QLSP. In particular, the proportionality in the encoding of $\boldsymbol{x}$ into the quantum state $|\boldsymbol{x}\rangle$ is linearly dependent on κ. Thus, the proportionality of $\langle \boldsymbol{x}|M|\boldsymbol{x}\rangle$ is quadratically dependent on κ, which increases the accuracy to which $|\boldsymbol{x}\rangle$ must be computed. We end this section by noting that the complexity of Algorithm 3 has no dependence on α_A. This is merely a consequence of the inequality $\kappa \geqslant \alpha_A \|A^{-1}\|$. Therefore, a block encoding of A with large α_A in the input is always accompanied by a proportionally large value of κ, thereby indirectly increasing the query- and 2-qubit gate cost of our algorithm.

V. DISCUSSION

We proved two main results in this paper. First, for the formulation of SLEP in which M is provided via block access, namely B-SLEP, we established a lower bound on queries to M. Second, we constructed a quantum algorithm for solving B-SLEP that saturates this lower bound, thereby proving that the lower bound is tight. We now discuss the implications of these two results.

We begin by comparing our lower bound with two previously known hardness results for restrictions of SLEP, as discussed in Sec. II C 1. The first of these results, presented in Theorem 3 [6], rules out the existence of a quantum algorithm solving SLEP with polylog$(N, 1/\epsilon)$ scaling of the total runtime. This result relies on no assumptions about how the input M is given, but instead requires the complexity-theoretic conjecture BQP $\neq$ PP. Our derivation of the lower bound does not make any complexity-theoretic assumption, but instead relies on the assumption that block access to M is given. Furthermore, our lower bound establishes a more stringent constraint $\Omega(1/\epsilon)$ on the scaling in ϵ of queries to M, thereby ruling out a sublinear scaling in ϵ. As state-of-the-art algorithms for generation of $|\boldsymbol{x}\rangle$ run in polylog$(N, 1/\epsilon)$ time, our result shows that the expectation-value-estimation step is exponentially harder with respect to scaling in ϵ than the $|\boldsymbol{x}\rangle$-generation step.

The second previously known hardness result, namely Theorem 2, states that if A is given by oracles, then no quantum algorithm for SLEP with poly$(\log N, \kappa)$ dependence can have query complexity in $O(1/\epsilon)$. This result only applies to those cases in which A is given by oracles, and does not apply, for instance, if A is fixed or if A is given by a description of a quantum circuit. In contrast, we establish our lower bound for B-SLEP by first deriving a lower bound for the case in which $A = \mathbb{1}_n$ is fixed. Therefore, our result shows that linear scaling on $1/\epsilon$ is optimal if either of the inputs A and M is given by oracles. SLEP for fixed A arises, for instance, in solving Poisson equation using the finite-element method with a fixed choice of basis function [39].

En route to deriving our lower bound for queries to B-SLEP, we derive a lower bound for S-EVHM (Problem 6), which is the problem of estimating the expectation value of a sparse-access Hermitian matrix M with respect to the state generated by a given unitary black box. Our lower bound for S-EVHM is $\Omega(d_M\beta_M/\epsilon)$, where d_M is the sparsity of M and β_M is an upper bound on the max-norm of M. As S-EVHM is equivalent to the restriction of SLEP to the case in which $A = \mathbb{1}_n$ is fixed and access to M is given by sparse-access oracles, the lower bound for S-EVHM also applies to SLEP in the sparse-access setting. We also derive a lower bound for B-EVHM (Problem 4), which is the problem of estimating the expectation value of a Hermitian matrix M provided by block access with respect to the state generated by a given unitary black box. Our lower bound for B-EVHM is $\Omega(\alpha_M/\epsilon)$, where α_M is the scaling factor for the block encoding of M. As $d_M\beta_M \geqslant \|M\|$ and $\alpha_M \geqslant \|M\|$, a common $\Omega(\|M\|/\epsilon)$ lower bound holds for estimating the expectation value of a black-box Hermitian matrix M irrespective of the oracle encoding of M. Our lower bound complements the lower bound on queries to V given by Corollary 4 in Ref. [34]. Interestingly, whereas our lower bound for B-EVHM holds under the restriction $V = \mathbb{1}_n$, our lower bound for S-EVHM is violated under the same restriction. As previous works did not consider the formulation of SLEP in which block access to M is given, the optimal scaling with respect to α_M was not known. Our lower bound for B-SLEP, which follows from our lower bound for B-EVHM, establishes a linear scaling in α_M.

To close, we discuss the implications of our algorithm for B-SLEP. Most importantly, the query complexity of our algorithm proves that our lower bound for queries to M for B-SLEP is tight up to a constant prefactor. Our algorithm for B-SLEP builds on the algorithm for B-EVHM in Ref. [24]. Our algorithm for B-SLEP also achieves linear scaling with respect to α_M for queries to M, which is optimal, and saturates the constraint on queries to A given by Theorem 2 up to logarithmic factors. Our algorithm for B-SLEP can be used to compute the expectation value of a smooth function of a Hermitian matrix with respect to the solution vector $\boldsymbol{x}$ by combining our algorithm with existing quantum algorithms for matrix arithmetic [22,23,40]. Finally, our algorithm for S-EVHM shows that our lower bound for the same problem is tight. This algorithm can be used as a subroutine along with Lemma 1 to construct an algorithm for SLEP in the sparse-access setting with optimal queries to M, although we do not detail this algorithm in our paper.

VI. CONCLUSION

Systems of linear equations arise in nearly all areas of science and engineering. The HHL algorithm and its improvements generate a quantum encoding of the solution vector $\boldsymbol{x}$ to a system of N linear equations $A\boldsymbol{x} = \boldsymbol{b}$ in cost polylog(N) [6]. This logarithmic dependence on N can be translated into efficient solutions for practical problems if the value $\boldsymbol{x}^\dagger M\boldsymbol{x}$ can be estimated for a given M in cost polylog(N). However, the cost of this estimation can be prohibitively expensive [15]. To address this caveat, we determine the complexity of the system of linear equations problem (SLEP), in which the task is to estimate $\boldsymbol{x}^\dagger M\boldsymbol{x}$ to an additive accuracy ϵ given A, M, and $\boldsymbol{b}$. We consider the setting in which M is given as a block encoding [11]. Block encoding is a common method of specifying matrix input to computational problems, and it also allows derivation of lower bounds on the query cost.

We analyze the cost of solving SLEP in terms of queries to the block encoding of M. Our main result is that any quantum algorithm for solving SLEP has a cost that is of order α_M/ϵ if M is provided as a block encoding, where α_M is the proportionality factor of the block encoding. As $\log(\alpha_M/\epsilon) + 1$ digits are sufficient to provide an estimate to additive accuracy ϵ, our results imply that the cost of solving SLEP scales exponentially with the number of digits to which $\boldsymbol{x}^\dagger M\boldsymbol{x}$ is estimated. We also prove that our bound on the cost is tight with respect to ϵ by constructing a quantum algorithm that saturates our bound. This bound for SLEP can be used in conjuction with previous hardness results, which bound the queries to the encoding of A with respect to ϵ [6]. To prove our lower bound for SLEP, we first prove a lower bound for the problem B-EVHM, in which the task is to estimate the expectation value of a block-encoded matrix M with respect to a given quantum state. To derive a lower bound for B-EVHM, we leverage a known lower bound for estimating the mean of a black-box function [30]. Our algorithm for SLEP relies on a known algorithm for B-EVHM [24].

Our results rigorously prove that if M is provided by a block encoding and if α_M/ϵ is superpolylogarithmic in N, then the cost of any quantum algorithm for solving SLEP is also superpolylogarithmic in N. Consequently, quantum algorithms that employ a block encoding of M show promise only for those applications for which an accuracy with $\alpha_M/\epsilon \in$ polylog(N) is sufficient. If the scaling of α_M/ϵ with respect to N is faster than polylog(N), then an efficient solution is unlikely unless M is further restricted. An interesting question in this direction is whether such an efficient solution is possible if constraints on the rank or the eigenvalue distribution of M are known, in addition to restrictions on A and $\boldsymbol{b}$.

ACKNOWLEDGMENTS

This project is supported by the Government of Alberta and by the Natural Sciences and Engineering Research Council of Canada (NSERC). A.A. acknowledges support through the Killam 2020 Postdoctoral Fellowship. This work was partially completed while R. R. Nerem was at the Quantum Algorithms Institute, where he was funded by a Mitacs Accelerate Scholarship.

APPENDIX A: LOWER BOUND FOR B-EVHM FOR FIXED V

In this Appendix, we show that the lower-bound B-EVHM given by Theorem 6 also applies to the restriction of B-EVHM in which V is a fixed input, and not a unitary black box. We first adapt a theorem which shows that block access to two matrices A and B can be used to construct block access to AB.

Lemma 14 (Reference [11]). Given n qubits, block access $(\alpha_A, a_A, 0, U_A)$ to $2^n \times 2^n$ Hermitian matrix A and block access $(\alpha_B, a_B, 0, U_B)$ to $2^n \times 2^n$ Hermitian matrix B, then block access $(\alpha_A\alpha_B, a_A + a_B, 0, U_{AB})$ to AB can be constructed such that U_{AB} uses only a single call to U_A and a single call to U_B.

Algorithm 4. Algorithm for sparse-access expectation value of a Hermitian matrix.

```
Input: $n$ qubits, sparse access $(d_M, \beta_M, O_{M_{\rm val}}, O_{M_{\rm loc}})$ to $M$, an accuracy $\epsilon > 0$, and a black-box unitary $V$
Output: an $\epsilon$-additive approximation $\tilde{u}$ to $u := \langle 0|V^\dagger MV|\psi\rangle$ with probability $\frac{2}{3}$
Procedure:
Construct block access $(d_M\beta_M, 2, \epsilon, U_M)$ to $M$.                    ▷ use Lemma 1.
Compute $\tilde{u}$ by executing Algorithm 2 with input $(n, (d_M\beta_M, 2, \epsilon/2, U_M), \epsilon/2, V)$.
return $\tilde{u}$.
```

We now show that Theorem 6 applies when to B-EVHM under any restriction that fixes V.

Lemma 15. Let $\{V_n \in \mathcal{U}(\mathscr{H}_2^{\otimes n}) :\ n \in \mathbb{Z}^+\}$ be a set of unitary operators. Then, any algorithm $\mathcal{Q}$ that solves all instances of B-EVHM with input $(n, (\alpha_M, a_M, 0, U_M), 1/\epsilon, V = V_n)$ makes $\Omega(\alpha_M/\epsilon)$ queries to the block encoding of M.

Proof. We first prove that given an algorithm $\mathcal{Q}$ as described in the Lemma 15, any instance of B-EVHM can be solved by making one query to $\mathcal{Q}$ and without making any additional uses of the block encoding of M. The desired lower bound then follows immediately from Theorem 6. Note that

$$U_F = (V_nV^\dagger \otimes |0^{a_M}\rangle\langle 0^{a_M}|)U_M(VV_n^\dagger \otimes |0^{a_M}\rangle\langle 0^{a_M}|) \tag{A1}$$

is a $(\alpha_M, a_M, 0)$ block encoding of $F = V_nV^\dagger MVV_n^\dagger$. This block encoding can be instructed by Lemma 14. Then, any instance $(n, (\alpha_M, a_M, 0, U_M), 1/\epsilon, V)$ of B-EVHM can be solved by making one query to $\mathcal{Q}$ with input $(n, (\alpha_M, a_M, 0, U_F), 1/\epsilon)$. The output r of this algorithm satisfies

$$|r - \langle 0^n|V_n^\dagger V_nV^\dagger MVV_n^\dagger V_n|0^n\rangle| < 2/3 \tag{A2}$$

which implies

$$|r - \langle 0^n|V^\dagger MV|0^n\rangle| < 2/3 \tag{A3}$$

with probability at least $\frac{2}{3}$ as required. ■

APPENDIX B: ALGORITHM FOR S-EVHM

In this Appendix, we construct an algorithm for S-EVHM to show that the query-complexity lower bound for S-EVHM given by Corollary 3 is tight.

Theorem 9. Given n qubits, sparse access $(d_M, \beta_M, O_{M_{\rm val}}, O_{M_{\rm loc}})$ to a $2^n \times 2^n$ Hermitian matrix M, an accuracy $\epsilon > 0$, and a unitary black box V, then an ϵ-additive approximation $\tilde{u}$ to $u := \langle 0|V^\dagger MV|0\rangle$ can be returned, with probability at least $\frac{2}{3}$, using $O(d_M\beta_M/\epsilon)$ queries to sparse-access oracles for M, $O(d_M\beta_M/\epsilon)$ queries to V as well as

$$O\left(\frac{d_M\beta_M}{\epsilon}\left(n + \log^{2.5}\left(\frac{d_M\beta_M}{\epsilon}\right)\right)\right), \tag{B1}$$

additional 2-qubit gates.

Proof. The algorithm (Alg. 4) is as follows:

Correctness. Let $\hat{M} := d_M\beta_M\langle 0^2|U_M|0^2\rangle$, so that $\|\hat{M} - M\| < \epsilon/2$. Then Algorithm 2 guarantees that

$$|\tilde{u} - \langle 0^n|V^\dagger\hat{M}V|0^n\rangle| < \epsilon/2, \tag{B2}$$

with probability at least $\frac{2}{3}$. We then have

$$\begin{aligned} |\tilde{u} - u| &\leqslant |\tilde{u} - \langle 0^n|V^\dagger\hat{M}V|0^n\rangle| \\ &\quad + |\langle 0^n|V^\dagger\hat{M}V|0^n\rangle - \langle 0^n|V^\dagger MV|0^n\rangle| \\ &\leqslant \epsilon/2 + \|\hat{M} - M\| < \epsilon, \end{aligned} \tag{B3}$$

with probability at least $\frac{2}{3}$ as required.

Complexity. The complexity of queries to sparse-access oracles follows directly from the complexities of Algorithm 2 with α_M set to $d_M\beta_M$. By Lemma 1, the block encoding U_M uses $O(n + \log^{2.5}(d_M\beta_M/\epsilon))$ additional 2-qubit gates. Furthermore, U_M is queried $O(d_M\beta_M/\epsilon)$ times by Algorithm 2. Therefore, our algorithm for S-EVHM uses the number of 2-qubit gates given in Eq. (B1). ■

The queries to the block encoding of M given in Theorem 5 saturate the lower bound for B-EVHM in Theorem 6. This theorem concludes the proof of the tightness of our lower bound for B-EVHM.

[1] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[2] D. W. Berry, High-order quantum algorithm for solving linear differential equations, J. Phys. A: Math. Theor. **47**, 105301 (2014).

[3] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[4] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[7] A. Ambainis, Variable time amplitude amplification and quantum algorithms for linear algebra problems, in *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, Leibniz International Proceedings in Informatics (LIPIcs), edited by C. Dürr and T. Wilke (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, 2012), Vol. 14, pp. 636–647, doi: 10.4230/LIPIcs.STACS.2012.636.

[8] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially

improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).

[9] B. Duan, J. Yuan, C.-H. Yu, J. Huang, and C. Hsieh, A survey on HHL algorithm: From theory to application in quantum machine learning, Phys. Lett. A **384**, 126595 (2020).

[10] J.-P. Liu, H. O. Kolden, H. Krovi, N. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, Proc. Natl. Acad. Sci. USA **118** (2021).

[11] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing* (Association for Computing Machinery, New York, 2019), doi: 10.1145/3313276.3316366.

[12] N.-H. Chia, A. Gilyén, T. Li, H. Lin, E. Tang, and C. Wang, Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning, in *Proceedings of the 52nd Annual ACM SIGACT Symposium on the Theory of Computing* (Association for Computing Machinery, New York, 2020), doi: 10.1145/3357713.3384314.

[13] G. Wang, Efficient quantum algorithms for analyzing large sparse electrical networks, Quantum Inf. Comput. **17**, 987 (2017).

[14] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum Fingerprinting, Phys. Rev. Lett. **87**, 167902 (2001).

[15] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291 (2015).

[16] D. Aharonov and A. Ta-Shma, Adiabatic quantum state generation and statistical zero knowledge, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03 (Association for Computing Machinery, New York, USA, 2003), pp. 20–29, doi: 10.1145/780542.780546.

[17] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. **270**, 359 (2007).

[18] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14 (Association for Computing Machinery, New York, 2014), pp. 283–292, doi: 10.1145/2591796.2591854.

[19] N. W. Andrew M. Childs, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[20] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, Simulating quantum dynamics on a quantum computer, J. Phys. A: Math. Theor. **44**, 445308 (2011).

[21] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, edited by V. Guruswami (IEEE Computer Society, Berkeley, CA, 2015), pp. 792–809, doi: 10.1109/FOCS.2015.54.

[22] G. H. Low and I. L. Chuang, Hamiltonian Simulation by Qubitization, Quantum **3**, 163 (2019).

[23] S. Chakraborty, A. Gilyén, and S. Jeffery, The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation, in *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, 2019), Vol. 132, pp. 33:1–33:14, doi: 10.4230/LIPIcs.ICALP.2019.33.

[24] P. Rall, Quantum algorithms for estimating physical quantities using block encodings, Phys. Rev. A **102**, 022408 (2020).

[25] R. P. Brent, Fast multiple-precision evaluation of elementary functions, J. ACM **23**, 242 (1976).

[26] S. Aaronson and G. Kuperberg, Quantum versus classical proofs and advice, in *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity (CCC'07)* (IEEE, Piscataway, NJ, 2007), pp. 115–128, doi: 10.1109/CCC.2007.27.

[27] A. S. Householder, Unitary triangularization of a nonsymmetric matrix, J. ACM **5**, 339 (1958).

[28] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, Quantum Inf. Comput. **12**, 29 (2012).

[29] G. Brassard and P. Bratley, *Fundamentals of Algorithmics* (Prentice-Hall, Hoboken, USA, 1996).

[30] A. Nayak and F. Wu, The quantum query complexity of approximating the median and related statistics, in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, STOC '99 (Association for Computing Machinery, New York, USA, 1999) pp. 384–393, doi: 10.1145/301250.301349.

[31] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, UK, 2010).

[32] E. Knill, G. Ortiz, and R. D. Somma, Optimal quantum measurements of expectation values of observables, Phys. Rev. A **75**, 012328 (2007).

[33] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, in *Quantum Computation and Information*, Contemporary Mathematics (American Mathematical Society, Washington DC, 2002), Vol. 305, doi: 10.1090/conm/305.

[34] W. J. Huggins, K. Wan, J. R. McClean, T. E. O'Brien, N. Wiebe, and R. Babbush, Nearly optimal quantum algorithm for estimating multiple expectation values, arXiv:2111.09283 (2021).

[35] D. Janzing and P. Wocjan, A simple PromiseBQP-complete matrix problem, Theory Comput. **3**, 61 (2007).

[36] W. M. Kirby and P. J. Love, Variational Quantum Eigensolvers for Sparse Hamiltonians, Phys. Rev. Lett. **127**, 110503 (2021).

[37] D. Aharonov, V. Jones, and Z. Landau, A polynomial quantum algorithm for approximating the jones polynomial, in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, STOC '06 (Association for Computing Machinery, New York, USA, 2006), pp. 427–436, doi: 10.1145/1132516.1132579.

[38] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, npj Quantum Inf. **7**, 52 (2021).

[39] A. Montanaro and S. Pallister, Quantum algorithms and the finite element method, Phys. Rev. A **93**, 032324 (2016).

[40] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter Error with Commutator Scaling, Phys. Rev. X **11**, 011020 (2021).