Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions

Yu Tong,[1] Dong An,[1] Nathan Wiebe,[2,3,4] and Lin Lin[5,6]

[1]*Department of Mathematics, University of California, Berkeley, California 94720, USA*
[2]*Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 2E4*
[3]*Department of Physics M5S 1A1, University of Washington, Seattle, Washington 98195, USA*
[4]*High Performance Computing Division, Pacific Northwest National Laboratory, Richland, Washington 99354, USA*
[5]*Department of Mathematics and Challenge Institute for Quantum Computation, University of California, Berkeley, California 94720, USA*
[6]*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

(Received 21 September 2020; revised 18 July 2021; accepted 23 August 2021; published 27 September 2021)

Preconditioning is the most widely used and effective way for treating ill-conditioned linear systems in the context of classical iterative linear system solvers. We introduce a quantum primitive called fast inversion, which can be used as a preconditioner for solving quantum linear systems. The key idea of fast inversion is to directly block encode a matrix inverse through a quantum circuit implementing the inversion of eigenvalues via classical arithmetics. We demonstrate the application of preconditioned linear system solvers for computing single-particle Green's functions of quantum many-body systems, which are widely used in quantum physics, chemistry, and materials science. We analyze the complexities in three scenarios: the Hubbard model, the quantum many-body Hamiltonian in the plane-wave-dual basis, and the Schwinger model. We also provide a method for performing Green's function calculation in second quantization within a fixed-particle manifold and note that this approach may be valuable for simulation more broadly. Aside from solving linear systems, fast inversion also allows us to develop fast algorithms for computing matrix functions, such as the efficient preparation of Gibbs states. We introduce two efficient approaches for such a task, based on the contour-integral formulation and the inverse transform, respectively.

DOI: 10.1103/PhysRevA.104.032422

I. INTRODUCTION

Linear systems appear ubiquitously in scientific and engineering computations. Accelerated solution of linear systems on quantum computers, or the quantum linear system problem (QLSP), has received a significant amount of attention in the past decade [1–12]. Solving QLSP means finding a solution vector $|x\rangle$ stored as a quantum state (up to a normalization constant), so that $|x\rangle = A^{-1}|b\rangle / \|A^{-1}|b\rangle\|$. The main advantage provided by a quantum computer is that the number of the qubits needed to store the matrix and the solution vector only scales logarithmically with respect to the matrix dimension, thus overcoming the curse of dimensionality on a classical computer. On the other hand, the cost of a quantum algorithm for solving a generic QLSP scales at least as $\Omega(\kappa(A))$ [8], where $\kappa(A) := \|A\|\|A^{-1}\|$ is the condition number of A.[1] This can be expensive if the linear system is ill conditioned. It is therefore of great interest if we can exploit certain special structures of QLSPs to reduce the cost.

For a classical iterative algorithm such as the conjugate-gradient (CG) method, the most effective way to accelerate the solution of ill-conditioned linear systems is to find a preconditioner M so that (1) $\kappa(MA) \ll \kappa(A)$, (2) the matrix-vector multiplication $M|\psi\rangle$ is easily accessible, and in particular its cost is independent of $\kappa(M)$ [13]. On a classical computer, the condition (2) can be satisfied, for instance, if M is a diagonal matrix or can be easily diagonalized, or if M is obtained by a sparse direct method such as the incomplete Cholesky factorization [14–16]. Then, the cost for solving the transformed equation $MA|x\rangle = M|b\rangle$ is determined by $\kappa(MA)$ instead of $\kappa(A)$. The same strategy can be used to reduce the complexity of a quantum linear solver [17].

In this paper, we focus on a QLSP of the form

$$|x\rangle = (A+B)^{-1}|b\rangle / \|(A+B)^{-1}|b\rangle\|, \tag{1}$$

where $A, B \in \mathbb{C}^{N\times N}$, and $N = 2^n$. Throughout the paper, unless stated otherwise, we assume $\|A\|$ can be very large, while $\|B\|$, $\|A^{-1}\|$, $\|(A+B)^{-1}\| = O(1)$. Therefore, the condition numbers $\kappa(A), \kappa(A+B) = \Theta(\|A\|)$. Such a scenario occurs frequently in scientific computing, e.g., if A is obtained by discretizing an unbounded operator, such as the Laplace operator in a confined domain (see Sec. II B), or if A represents a term in a quantum many-body Hamiltonian that is significantly larger than other terms (see Sec. IV C). We would like to obtain a quantum linear system solver, of which the cost is independent of $\|A\|$. In particular, we will illustrate in detail the computation of single-particle Green's functions of a quantum many-body system. It is worth pointing out that

[1]Throughout the paper $\|A\| \equiv \|A\|_2$ is the 2-norm (or the operator norm) of an operator A, and $\|u\| \equiv \|u\|_2$ is the 2-norm of a vector u.

although many efforts have been made to efficiently solve the QLSP, suitable applications of QLSP solvers remain scarce, as it is often difficult to efficiently load classical data into the quantum circuit, and to output useful information using a limited number of measurements. We demonstrate that the problem of computing Green's functions does not not suffer from such problems, and can be a suitable end-to-end application of QLSP solvers.

A closely related, and more general, problem is to evaluate matrix functions of the form $f(A+B)$, where $f(\cdot)$ is a smooth function defined on the spectrum of $H=A+B$ [18]. Here for simplicity we assume A, B are Hermitian matrices, so the spectrum of H is on the real line. Under similar assumptions above, we would like to obtain quantum algorithms of which the cost is independent of $\|A\|$. This obviously depends on the choice of the function f. We will focus, for concreteness, on the function $f(x)=e^{-x}$. This choice is motivated in part by the importance of preparing Gibbs states in quantum simulation [19], machine learning [20], and quantum algorithms for semidefinite programming [21].

A. Overview of the results and related works

Quantum linear system solver. Starting from the groundbreaking work of [8], in the past decade, several quantum algorithms have been developed to improve the performance of generic QLSP solvers [1,2,6,7,9,10]. We review these methods in Appendix A.

Recently, quantum-inspired classical algorithms based on certain ℓ^2-norm sampling assumptions [22,23] have been developed that are only up to polynomially slower than quantum linear system solvers. However, it is unclear whether the classical ℓ^2-norm sampling can be achieved efficiently without access to a quantum computer in the setting of this work. The quantum-inspired classical algorithms also suffer from many practical issues making their application limited to highly specialized problems [24]. Most importantly, the assumption of low rankness is crucial in these algorithms. The methods presented in this work assume a block-encoding model, which could be used to efficiently represent low-rank as well as full-rank matrices on a quantum computer.

Preconditioned quantum linear system solver. Solving the quantum linear system problem using preconditioning to deal with large condition number has been discussed in works such as [17,25]. The idea of [17] is to use the sparse approximate inverse (SPAI) preconditioner [26,27], which uses a d-sparse matrix as a preconditioner. The SPAI, denoted by M, can be constructed by solving a least-squares procedure for each row of the matrix A. However, for many problems, an efficient preconditioner in the form of SPAI may not exist, and the work of [17] did not provide an efficient quantum implementation to construct SPAI nor its performance analysis. In [25], the preconditioner M is taken to be a circulant matrix that can be efficiently diagonalized using a quantum Fourier transform (QFT). However, the complexity of the algorithm in [25] can depend on $\kappa(M)$, which should not be expected to be smaller than $\kappa(A)$. Furthermore, neither work provides an upper bound for $\kappa(MA)$, which is a key quantity determining, e.g., the circuit depth.

In this paper, we propose a different mechanism for constructing efficient preconditioners, called *fast inversion*. The inspiration of fast inversion is that any invertible, 1-sparse matrix A can be efficiently implemented on a quantum computer via classical arithmetics. In particular, the cost of constructing a block encoding of A^{-1} is independent of $\kappa(A)$. Note that fast inversion does not violate the complexity lower bound for solving QLSP, which is a statement of the efficiency of QLSP solvers [8] applied to general matrices. This is in parallel to the fast-forwarding process of 1-sparse matrices in Hamiltonian simulation [28,29], which does not violate the theorem of "no-fast-forwarding" [30]. Furthermore, if A can be unitarily diagonalized, so that both the diagonalization procedure and the encoding of eigenvalues can be efficiently implemented on a quantum computer, then A can be fast inverted. Fast inversion can be viewed as a quantum primitive for a wide range of tasks. For example, we describe an efficient implementation of inverting certain normal matrices, such as circulant matrices.

We introduce a parameter $\xi = \|A^{-1}|b\rangle\|$, and without loss of generality rescale A so that $\|A^{-1}\|=1$. As will be analyzed in Sec. II, if we consider ξ and $\kappa(A)$ as two independent parameters, the immediate benefit of the fast inversion is that unlike any other methods in the literature, the cost of solving the QLSP depends only on ξ. The value of ξ is bounded from below by $1/\kappa(A)$. Therefore, in the worst case when $\xi = 1/\kappa(A)$, the cost for solving the QLSP still depends linearly on $\kappa(A)$. However, we will demonstrate in Remark 5 that such dependence through ξ already reaches the complexity lower bound, and cannot be improved by any QLSP solver. Furthermore, such a bound for ξ is usually not tight for many examples of practical interest. This is demonstrated via a concrete example of using fast inversion to solve a translational-invariant elliptic partial differential equation (PDE) in Sec. II B.

Now for the linear system (1), assuming A can be fast inverted so that $M=A^{-1}$, the cost for solving the the preconditioned linear system of Eq. (1) depends on $\kappa(M(A+B)) = \kappa(I+A^{-1}B)$, which can be bounded in terms of $\|B\|, \|A^{-1}\|, \|(A+B)^{-1}\|$ (or, more accurately, their block-encoding factors, see Lemma 1). This is in contrast to other QLSP preconditioning techniques where no such bound is known. We introduce a parameter $\xi = \|(A+B)^{-1}|b\rangle\|$ where $|b\rangle$ is the normalized quantum state corresponding to the right-hand side. Using quantum singular value thresholding (QSVT), we obtain a gate-based implementation of a preconditioned linear system solver, and its cost is independent of $\|A\|$, but only depends on ξ, and several block-encoding subnormalization factors which are upper bounds of the norms $\|B\|, \|A^{-1}\|$, and $\|(A+B)^{-1}\|$ (Theorem 1 and Corollary 1). In the worst case the query complexity of the preconditioned linear system solver can depend superlinearly on $\|B\|$ (or the corresponding block-encoding factor α_B). However, in some cases the worst case estimate can be significantly improved and the scaling with respect to $\|B\|$ can be linear. We discuss such implications in Remarks 6 and 7. Throughout the paper we adopt the worst-case estimate of $\|(I+A^{-1}B)^{-1}\|$, which is responsible for the superlinear scaling with respect to α_B in Tables I and II below.

Computing single-particle Green's functions of quantum many-body systems. As an application of the preconditioned

TABLE I. Comparison of the number of queries to the ground state and relevant block encodings needed using different algorithms for computing an entry of the single-particle Green's function $G(z)$ of a quantum many-body Hamiltonian of the form $\hat{H} = \hat{A} + \hat{B}$, where $|z| \geqslant \eta$. The operators $\hat{A}$, $\hat{B}$, and $\hat{H}$ are given in their block encodings with subnormalization factors α_A, α_B, and α_H, respectively, satisfying $\alpha_H \sim \alpha_A \sim \|\hat{A}\| \gg \alpha_B$. $\widetilde{\sigma}_{\min}$ is defined in Theorem 2 and $\widetilde{\sigma}_{\min} = \Omega(\eta/\alpha_B)$. The error comes from three parts: preparing the ground state, block encoding of the matrix inverse, and the Hadamard test with amplitude estimation. The latter two are controlled by ϵ while the first is controlled by ς. For simplicity we assume the ground energy is known exactly here. The error as a result of inexact ground energy is included in Theorem 2.

	Algorithm	Queries to U_Ψ	Queries to block encodings	Error
w.o. preconditioner	HHL	$O\left(\frac{1}{\eta\sqrt{p}\epsilon}\ln\left(\frac{1}{\varsigma}\right)\right)$	$\widetilde{O}\left(\frac{\lvert z\rvert+\alpha_H}{\eta^3\epsilon^2}\right)$	$\epsilon + O\left(\frac{\varsigma}{\eta}\right)$
	LCU/QSVT	$O\left(\frac{1}{\eta\sqrt{p}\epsilon}\ln\left(\frac{1}{\varsigma}\right)\right)$	$\widetilde{O}\left(\frac{\lvert z\rvert+\alpha_H}{\eta^2\epsilon}\right)$	$\epsilon + O\left(\frac{\varsigma}{\eta}\right)$
w. preconditioner	This work	$O\left(\frac{1}{\widetilde{\sigma}_{\min}\sqrt{p}\epsilon}\ln\left(\frac{1}{\varsigma}\right)\right)$	$\widetilde{O}\left(\frac{\alpha_B}{\widetilde{\sigma}^2_{\min}\epsilon}\right)$	$\epsilon + O\left(\frac{\varsigma}{\widetilde{\sigma}_{\min}}\right)$

linear system solver, we consider the problem of computing the one-particle Green's function of a quantum many-body system, which is a standalone linear system problem in high dimensions. Most of the literature on quantum simulation thus far focus on estimating the ground-state energy and preparing the ground state. However, once the ground state is found, one may further evaluate the single-particle Green's function, which carries important spectroscopic information and is widely used in quantum physics, chemistry, and materials science [31,32]. Calculation of Green's functions is computationally challenging. Previous works [33,34] focused on evaluating Green's functions in the time domain via Hamiltonian simulation. Reference [35] computes the response function, which is closely related to the Green's function, in the frequency domain. Here we will provide a preconditioned linear system method for direct computation of the Green's function in the frequency domain.

The setup of the problem is as follows. Suppose we are given a Hamiltonian $\hat{H} = \hat{A} + \hat{B}$, so that $z\hat{I} + \hat{A}$ can be fast inverted for some properly chosen $z \in \mathbb{C}$. We assume we have an $(\alpha_H, m_H, 0)$ block encoding of $\hat{H}$, as well as the oracles introduced in Sec. IV B. We also assume there is an oracle available to construct the ground state $|\Psi_0\rangle$ of the Hamiltonian to precision ς in terms of trace-norm distance with probability at least p, and we denote this oracle by U_Ψ. The goal is to compute the Green's function $G^{(+)}_{ij}(z) = \langle\Psi_0|\hat{a}_i(z - \hat{H} + E_0)^{-1}\hat{a}^\dagger_j|\Psi_0\rangle$ as defined in Sec. IV A (and a corresponding $G^{(-)}$) for z satisfying $|\operatorname{Im} z| \geqslant \eta > 0$.

This task can be accomplished using either Harrow-Hassidim-Lloyd (HHL) (based on phase estimation), linear combination of unitaries (LCU), or quantum singular value transformation (QSVT), to construct a block encoding of $(z - \hat{H} + E_0)^{-1}$. We then apply the nonunitary Hadamard test described in Appendix D to estimate the expectation value. The analysis in Appendix F shows that the number of queries to U_Ψ scales linearly with the block-encoding subnormalization factor of $(z - \hat{H} + E_0)^{-1}$, which is upper bounded by $1/\eta$, and the number of queries to the block encoding of $\hat{H}$ scales linearly with the product of the above subnormalization factor and the number of queries used in the block encoding of the matrix inverse, with the latter scaling linearly with the condition number. Here the condition number scales linearly with $|z| + \alpha_H \sim |z| + \|\hat{A}\|$. Using our preconditioning technique we can remove this dependence on $z + \|\hat{A}\|$. The detailed analysis can be found in Sec. IV B. We also provide a few concrete examples such as the Hubbard model, the quantum many-body Hamiltonian in a plane-wave dual basis set, and the Schwinger model in Sec. IV C.

In this application, the outputs are the matrix elements of the Green's function $G_{ij}(z)$, rather than a quantum state representing the solution to a QLSP. Therefore, the complexity does not involve the parameter ξ as in the setting of the preconditioned QLSP. As a consequence, the speedup we discussed in the previous paragraph depends only on the structure of the Hamiltonian.

Fast algorithm for evaluating matrix functions. The method of solving QLSP (1) is a special case of computing matrix functions $f(A + B)|b\rangle$ with $f(x) = x^{-1}$. Here for simplicity we restrict A, B to be Hermitian matrices. In parallel to solving QLSP, the evaluation of a general matrix function can also be performed using the phase estimation algorithm, similar to its use in the HHL algorithm [8]. Similarly, LCU [6,36], and QSP/QSVT [7,37] can also be used to evaluate matrix functions, and achieve better dependence on various parameters, especially the desired precision ϵ.

TABLE II. Comparison of the performance of different algorithms for preparing the state $e^{-H}|b\rangle/\xi$, where $\xi = \|e^{-H}|b\rangle\|$ and $H = A + B \succ 0$. We assume A, B, and H are given in their block encodings with subnormalization factors α_A, α_B, and α_H, respectively, and $\alpha_H \sim \alpha_A \sim \|A\| \gg \alpha_B$. $\widetilde{\sigma}'_{\min}$ and $\widetilde{\sigma}_{\min}$ are defined in Theorems 3 and 6, respectively. $\widetilde{\sigma}'_{\min} = \Omega(1/\alpha_B)$ and $\widetilde{\sigma}_{\min} = \Omega[1/(1 + \|(A + B)^{-1}\|\|B\|)]$. Reference [38], which uses phase estimation to prepare Gibbs state, estimates the number of queries to time evolution, instead of block encodings of the Hamiltonians. In this table we assume the time evolution is done using Hamiltonian simulation methods such as in Ref. [40], which simulates time evolution for time t with $\widetilde{O}(\alpha_H t)$ queries to U_H. It also uses $O(\ln(\frac{1}{\xi\epsilon}))$ qubits in the "energy register."

	Algorithm	Query complexities
w.o. preconditioner	Phase estimation [38]	$\widetilde{O}\left(\frac{\alpha_H}{\xi\epsilon}\right)$
	LCU [39]	$\widetilde{O}\left(\frac{\alpha_H}{\xi}\ln\left(\frac{1}{\epsilon}\right)\right)$
w. preconditioner	This work	$\widetilde{O}\left(\frac{\alpha_B}{\xi\widetilde{\sigma}^2_{\min}}\ln\left(\frac{1}{\epsilon}\right)\right)$
	(contour integral)	w
	This work	$\widetilde{O}\left(\frac{\alpha_B}{\xi\widetilde{\sigma}^2_{\min}}\left[\ln\left(\frac{1}{\epsilon}\right)\right]^5\right)$
	(inverse transformation)	

The cost of each method depends on the actual form of $f(\cdot)$. Here for concreteness we consider $f(x) = e^{-x}$, which is directly related to the problem of preparing Gibbs state s in quantum physics. Without loss of generality we assume $H = A + B \succ 0$, so that $|f| \leqslant 1$ evaluated on the spectrum of H. The costs of preparing the state $e^{-H}|b\rangle/\xi$ where $\xi = \|e^{-H}|b\rangle\|$, using phase estimation and LCU, are given in Table II, and they all depend directly on the subnormalization factor in the block encoding of H denoted by α_H. Naturally we have $\alpha_H \geqslant \|H\| \sim \|A\|$. We note that the Gibbs state preparation is a special case of the task discussed above. We can simply set $|b\rangle$ to be the maximally entangled state to obtain a purified Gibbs state. In this particular case $\xi = \sqrt{N/Z}$ where N is the Hilbert space dimension and $Z = \mathrm{Tr}(e^{-H})$ is the partition function. This will be discussed in Sec. V C. For simplicity and in order to be consistent with other works such as Ref. [39] we omit the dependence on temperature in the Table II, and this dependence will be discussed in Sec. V C as well.

We propose two methods to evaluate e^{-H} given the ability of fast inversion of A. The first method is based on the Cauchy contour-integral formulation

$$f(x) = \frac{1}{2\pi i}\oint_{\mathcal{C}} \frac{f(z)}{z-x}dz, \tag{2}$$

where $\mathcal{C}$ is a simple closed curve and $f(\cdot)$ is analytic on a region containing $\mathcal{C}$ and its interior, and x is inside $\mathcal{C}$. After proper discretization, the evaluation of the matrix function $f(A+B)$ becomes solving a series of preconditioned quantum linear system problems, which can be combined together using LCU. Our second method is based on a simple inverse transformation, namely,

$$f(x) = f(y^{-1} - a), \tag{3}$$

where $y = (x+a)^{-1}$ for some $a \in \mathbb{C}$. When $A+B$ is invertible, we may simply take $a = 0$. Again we use preconditioned quantum linear system solver, combined with a standard LCU/QSVT procedure, to evaluate $f(A+B)$. The cost of the two methods to prepare $e^{-(A+B)}$ is given in Table II, which is independent of $\|A\|$.

There is a class of algorithms based on quantum walks and Metropolis sampling to prepare the Gibbs state [41–43], which can be seen as a special case of implementing the matrix function e^{-H}. The complexity typically depends mainly on the gap of the transition matrix of the Markov chain, and thus the complexity estimate involves a different set of parameters. Therefore, we do not compare the complexities of these algorithms with our methods. For example, Ref. [44] uses LCU to prepare the Gibbs state, but it uses a different input model of the Hamiltonian from the block encoding we use in this work.

B. Notations

A matrix $A \in \mathbb{C}^{2^n \times 2^n}$ is referred to as an n-qubit matrix or n-qubit operator. Unless otherwise explained, we use the notation $N = 2^n$, and $[N] = \{0, \ldots, N-1\}$. We will extensively use the technique of block encoding, which is a way of embedding an arbitrary matrix as a submatrix of a larger unitary matrix. Here using a unitary matrix U_A to encode A as a submatrix means that there exists a normalizing constant $\alpha > 0$ such that

$$U_A = \begin{bmatrix} A/\alpha & * \\ * & * \end{bmatrix}, \tag{4}$$

where $*$ denotes arbitrary matrix blocks of proper sizes. In general, the matrix that we block encode may only approximate A/α. We use the following notation to describe such encodings.

Definition 1 (Block encoding [45]). An $(m+n)$-qubit unitary operator U_A is called an (α, m, ϵ) block encoding of an n-qubit operator A, if $\|A - \alpha(\langle 0^m| \otimes I_n)U_A(|0^m\rangle \otimes I_n)\| \leqslant \epsilon$.

Here m is the number of ancilla qubits for block encoding, and α is called the block-encoding factor, or the subnormalization factor. The block encoding has long been explicitly used in algorithms such as the quantum linear systems algorithm [8]. The block encoding is a powerful and versatile model, which can be used to efficiently encode density operators, Gram matrices, positive-operator valued measure (POVM), sparse-access matrices, as well as addition and multiplication of block-encoded matrices (we refer to [45] for a detailed illustration of such constructions).

Remark 1. For simplicity of discussion, we may often assume the given block encodings are error free, e.g., we may assume U_A is an $(\alpha, m, 0)$ block encoding of A. The error due to the given block encodings can often be taken into account without much technical difficulties, but may complicate the presentation of results. We can then focus on the error introduced by other parts of the algorithm, such as that due to the polynomial approximation of smooth functions.

We also use the following notations throughout the paper: The block encoding of a matrix A is generally denoted by U_A. Since the 2-norm of a unitary matrix U_A is 1, it is guaranteed that the 2-norm of A/α, which is a submatrix of U_A, is upper bounded by 1. This implies $\|A\| \leqslant \alpha$. Therefore, in this work we usually bound the norm of a matrix in terms of its block-encoding subnormalization factor, which in many cases is known *a priori*, for example, in the case of d-sparse matrices [6,45]. Since this paper uses the inverse of matrices extensively, we may use $U'_A := U_{A^{-1}}$ to denote the block encoding of A^{-1}. For convenience, we use α_A, m_A to denote the subnormalization factor and the number of ancilla qubits for A, respectively, and use α'_A, m'_A to denote those for A^{-1}. Throughout the paper we also frequently use $\widetilde{\sigma}_{\min}$ to denote a lower bound of the smallest singular value of a matrix of the form $I + A^{-1}B$.

To simplify the notation, we may omit the normalization factor in the QLSP problem $|x\rangle = A^{-1}|b\rangle/\|A^{-1}|b\rangle\|$, and write $|x\rangle \propto A^{-1}|b\rangle$ or $A|x\rangle \propto |b\rangle$. However, the normalization factor is not arbitrarily chosen, and the resulting state $|x\rangle$ is well defined. Although the phase factor in $|x\rangle$ is often not important, this allows us to define the distance between an approximate solution to QLSP $|\widetilde{x}\rangle$ and the true solution directly via the vector 2-norm $\|\,|\widetilde{x}\rangle - |x\rangle\,\|$. In this paper we mostly use the vector 2-norm to quantify error, with the exception in Theorem 2 where we use trace distance to quantify the error of ground-state preparation. When we say a target quantum state $|\phi\rangle$ is prepared to precision ϵ, it means that we prepare a quantum state $|\psi\rangle$ such that $\|\,|\phi\rangle - |\psi\rangle\,\| \leqslant \epsilon$. The relationship between the 2-norm distance and the more commonly used

fidelity and trace distance is as follows: if for two pure states $|\phi\rangle$ and $|\psi\rangle$, $\| |\phi\rangle - |\psi\rangle \| = \epsilon$, then the fidelity F satisfies

$$F := |\langle\phi|\psi\rangle|^2 \geqslant (1-\epsilon^2/2)^2,$$

and the trace distance between the two density matrices $\rho_\phi = |\phi\rangle\langle\phi|$ and $\rho_\psi = |\psi\rangle\langle\psi|$ satisfies

$$\begin{aligned} T(\rho_\phi, \rho_\psi) &:= \frac{1}{2}\operatorname{Tr}\left[\sqrt{(\rho_\phi-\rho_\psi)^\dagger(\rho_\phi-\rho_\psi)}\right] \\ &= \sqrt{1-|\langle\phi|\psi\rangle|^2} \leqslant \sqrt{1-(1-\epsilon^2/2)^2} \\ &= \epsilon\sqrt{1-\epsilon^2/4}. \end{aligned}$$

Additionally, we use the following asymptotic notations aside from the usual O notation throughout the paper: we write $f = \Omega(g)$ if $g = O(f)$; $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$; $f = \widetilde{O}(g)$ if $f = O(g \operatorname{polylog}(g))$. We denote by $C^m(\mathcal{I})$ set of functions on an interval $\mathcal{I}$, which is differentiable m times and the mth derivative is continuous. Correspondingly, $C^\infty(\mathcal{I})$ is the set of infinitely differentiable functions on $\mathcal{I}$ (also called smooth functions).

Remark 2. (Dilation of a non-Hermitian matrix). When solving QLSP, it is often assumed that A is a Hermitian matrix [6,8]. This is because a non-Hermitian matrix can be dilated into a Hermitian matrix using one ancilla qubit

$$\widetilde{A} = \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix}. \tag{5}$$

When A is given by its block encoding U_A, the dilated Hermitian matrix $\widetilde{A}$ can be obtained through $U_{\widetilde{A}} = |0\rangle\langle 1| \otimes U_A + |1\rangle\langle 0| \otimes U_A^\dagger$ with subnormalization factor 1. Note that this requires the controlled version of $U_A, U_A^\dagger$. The quantum singular value transformation technique in Appendix B can directly solve QLSP for non-Hermitian matrices without requiring a dilation step. In this paper, we assume $A \in \mathbb{C}^{N\times N}$ is a general square matrix, and will explicitly specify when A is taken to be a Hermitian matrix.

C. Organization of this paper

The rest of the paper is organized as follows. In Sec. II we discuss certain matrices we can fast invert on a quantum computer. This enables us to precondition linear systems, which is discussed in Sec. III. We then discuss two applications of the preconditioning technique we developed: computing the many-body Green's function in Sec. IV, and evaluating matrix function $e^{-\beta H}$ in Sec. V. Conclusion and discussion are given in Sec. VI. A brief review of quantum singular value transformation for solving QLSP, together with certain details of proofs and constructions, is given in the Appendices.

II. FAST INVERSION

Our preconditioning method relies on fast inverting a certain class of matrices efficiently.

Definition 2 (Fast-invertible matrices). A matrix A is fast invertible if, after rescaling A so that $\|A^{-1}\| = 1$, a $(\Theta(1), m, \epsilon)$ block encoding of A^{-1} can be obtained, and the number of queries to the oracles that determine A is independent of the condition number $\kappa(A)$.

In this definition, the oracles are not restricted to yield a direct block encoding of A. This can be seen in the examples of the unitarily diagonalizable matrices (Sec. II B) and the 1-sparse matrices (Sec. II A and Appendix C).

Before further discussion, we first clarify the relation between the notion of fast-invertible matrices we propose here and the notion of fast-forwardable matrices. The two concepts are clearly closely related. In particular, if A is a nonsingular, Hermitian matrix A that can be unitarily diagonalized efficiently, then A is both fast invertible and fast forwardable, in the sense that the circuit depth for constructing the block encoding of A^{-1} and e^{iAt} can be independent of $\kappa(A)$ and t, respectively. However, there is also an important difference: if A is fast forwardable, then the query complexity for preparing the state $e^{iAt}|b\rangle$ can be independent of t for any $|b\rangle$. On the other hand, if a matrix A is fast invertible, then to prepare a normalized state that is parallel to $A^{-1}|b\rangle$, the query complexity still depends on $\|A^{-1}|b\rangle\|$, which in the worst case is lower bounded by $1/\kappa(A)$, if we rescale A so that $\|A^{-1}\| = 1$. However, this lower bound is often not tight, as can be seen in the d-dimensional elliptic PDE example we discuss in Proposition 3, and leads to vast overestimation of the cost. Therefore, we take $\|A^{-1}|b\rangle\|$ as an independent parameter rather than using the worst-case bound $1/\kappa(A)$. There are also instances in which the goal is not to prepare a quantum state but to read out a scalar value, as in Green's-function evaluation discussed in Sec. IV. For these instances $\|A^{-1}|b\rangle\|$ can be irrelevant and the number of queries to A can be completely independent of $\kappa(A)$.

A. Fast inversion of diagonal and general 1-sparse matrices

We first consider a diagonal matrix $D \in \mathbb{C}^{N\times N}$. Note that $\|D\| = \max_i |D_{ii}|$, and $\|D^{-1}\| = (\min_i |D_{ii}|)^{-1}$. The condition number is then $\kappa(D) = \max_i |D_{ii}| / \min_i |D_{ii}|$. Our goal is to solve the QLSP, i.e., to obtain a normalized state

$$|x\rangle \propto D^{-1}|b\rangle, \tag{6}$$

where

$$|b\rangle = \sum_{i\in[N]} b_i|i\rangle.$$

Now assume that the diagonal entry of D is accessible via an oracle

$$O_D|i\rangle|0^l\rangle = |i\rangle|D_{ii}\rangle, \quad i \in [N], \tag{7}$$

where $|D_{ii}\rangle$ is a binary representation of the diagonal entry D_{ii} (for simplicity assume the l-bit approximation of D_{ii} is exact).

Remark 3. We assume each diagonal entry of D can be efficiently computed using a classical Boolean circuit of size $O(\operatorname{polylog}(N))$ with $m = O(\operatorname{polylog}(N))$ ancilla bits. In this case we can construct a quantum circuit with $O(\operatorname{polylog}(N) + l)$ gates and $O(\operatorname{polylog}(N) + l)$ ancilla qubits to implement this oracle O_D [46, Lemma 10.10]. For simplicity we omit these ancilla qubits in Eq. (7) since after reversing these quantum gates, their values will return to $|0^m\rangle$ at the end of the computation.

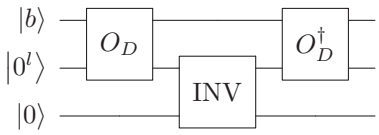


FIG. 1. Quantum circuit for fast inversion.

The circuit for the block encoding of D^{-1}, denoted by $U_{D^{-1}}$, is as in Fig. 1. Here the inversion circuit INV satisfies

$$\mathrm{INV}|\zeta\rangle|0\rangle = |\zeta\rangle\left(\frac{1}{\alpha'_D\zeta}|0\rangle + \sqrt{1-\left|\frac{1}{\alpha'_D\zeta}\right|^2}|1\rangle\right), \quad (8)$$

where $|\zeta\rangle$ is an l-qubit state representing an l-bit number ζ. This circuit is also used in the HHL algorithm [8]. Here $\alpha'_D \geqslant (\min_i |D_{ii}|)^{-1} = \|D^{-1}\|$. The output of the circuit is

$$U'_D|b\rangle|0^l\rangle|0\rangle = \sum_i (\alpha'_D D_{ii})^{-1} b_i|i\rangle|0^l\rangle|0\rangle + \sum_i \sqrt{1-|(\alpha'_D D_{ii})^{-1}|^2}\, b_i|i\rangle|0^l\rangle|1\rangle.$$

Running the circuit and measuring the ancilla registers (i.e., the last two registers), and upon getting all zero output, which we take as a success, we will have a quantum state proportional to $D^{-1}|b\rangle$ in the first register. Hence, U'_D is an $(\alpha'_D, m'_D, 0)$ block encoding of D^{-1} (recall $U'_D \equiv U_{D^{-1}}$) and $m'_D = O(l + \mathrm{polylog}(N))$ when we take into account the ancilla qubits that have been omitted as mentioned in Remark 3.

Proposition 1 (Fast inversion for diagonal matrices). For a diagonal matrix D whose diagonal entries can be accessed through the oracle O_D given in Eq. (7), and $\alpha'_D \geqslant 1/\min_i\{|D_{ii}|\}$, we can construct an $(\alpha'_D, m'_D, 0)$ block encoding of D^{-1}, given in Fig. 1, using O_D and $O_D^\dagger$ both exactly once. Here $m'_D = O(l + \mathrm{polylog}(N))$ if O_D uses $O(l + \mathrm{polylog}(N))$ ancilla qubits.

For simplicity of discussion here, we assume $\alpha'_D = \|D^{-1}\|$. The success probability of the above procedure is $(\|D^{-1}|b\rangle\|/\alpha'_D)^2$. Denote by $\xi = \|D^{-1}|b\rangle\|$ (consistent with that in Appendix B). With amplitude amplification [47] we can boost the success probability to be greater than $\frac{1}{2}$ by repeating the above process $O(\alpha'_D/\xi)$ times. Hence, the success probability depends on both the operator D as well as the state $|b\rangle$. In the worst case $\langle i|b\rangle$ vanishes everywhere else other than $i = \arg\max_k |D_{kk}|$. Then $\xi = \|D\|^{-1}$, and the number of repetitions becomes $O(\|D^{-1}\|\|D\|) = O(\kappa(D))$. On the other hand, if $|\langle j|b\rangle|$ is lower bounded by a constant for $j = \arg\min_i |D_{ii}|$, then $\xi = \Theta(\|D^{-1}\|)$, and the success probability of the fast inversion is $(\xi/\alpha'_D)^2 = \Omega(1)$. In this case, the cost of the fast inversion is $O(1)$ and is independent of $\kappa(D)$.

The cost of a QLSP solver may be reduced, if the effective condition number is much smaller than the condition number. Here the effective condition number refers to the ratio between the largest and the smallest singular values, whose corresponding singular vectors have a nonzero overlap with the right-hand side $|b\rangle$. Note that this is *not* the reason of the speedup we discuss above. In the above discussion we do not require $\langle j|b\rangle$ to vanish anywhere, and therefore our method does not rely on the effective condition number being smaller than $\kappa(D)$.

Let us now contrast the results above with the standard QSVT method for solving linear systems (briefly reviewed in Appendix B). Start from an $(\alpha_D, m_D, 0)$ block encoding of D denoted by U_D, we may take $\alpha_D = \|D\|$. Applying Theorem 7, using QSVT we can solve the QLSP (6) and obtain the solution to precision ϵ with probability at least $\frac{1}{2}$, using $O((\kappa(D)^2/\|D\|\xi)\ln[\kappa(D)/(\|D\|\xi\epsilon)])$ queries to U_D and $U_D^\dagger$. The circuit depth for block encoding the matrix inversion is $O(\kappa(D)\ln[\kappa(D)/(\|D\|\xi\epsilon)])$, and this circuit and its inverse are repeated $O(\kappa(D)/\|D\|\xi)$ times in the amplitude amplification [47] procedure. So the fast inversion is always more efficient in terms of the circuit depth for block encoding the matrix inversion. Considering the entire procedure for solving the QLSP, we need to take the value of ξ into account. Since $\xi \in [1/\|D\|, \|D^{-1}\|]$, in the worst case when $\xi = 1/\|D\|$, the fast-inversion method results in a quadratic speedup with respect to $\kappa(D)$. In the best case when $\xi = \|D^{-1}\| = \kappa(D)/\|D\|$, the cost of QSVT is still $\widetilde{O}(\kappa)$ while the cost of the fast inversion is $O(1)$.

To illustrate how fast inversion works, let us consider a concrete example:

$$D = \sum_{j=1}^{n} Z_j + (n+1)I_n, \quad (9)$$

where Z_j is the Pauli-Z matrix on the jth qubit, and I_n is the n-qubit identity operator. Then $\|D\| = 2n+1$, $\|D^{-1}\| = 1$, and $\kappa(D) = 2n+1$. We may construct a $(1, m, \epsilon)$ block encoding of D^{-1} as follows. Given a state $|i\rangle \equiv |s_1 \ldots s_n\rangle$ with $i \in [N]$ represented by a binary string and $s_j \in \{0, 1\}$, we may first use take O_D to be a quantum adder circuit, i.e.,

$$O_D|i\rangle|0^l\rangle = |i\rangle|D_{ii}\rangle, \quad D_{ii} = \left(2\sum_{j=1}^{n} s_j\right) + 1,$$

which can be implemented using a quantum adder circuit that uses $l = \lceil \ln n \rceil + 2$ ancilla qubits. We then find a $(1, l+1, 0)$ block encoding of U'_D as in Fig. 1. If the right-hand side vector $|b\rangle = |0^n\rangle$, then the number of repetitions needed to achieve $\Omega(1)$ success probability is $O(\kappa(D)) = O(n)$. On the other hand, if $|b\rangle = |1^n\rangle$, only $O(1)$ repetitions are sufficient for the fast-inversion method to succeed.

For 1-sparse matrices that are not necessarily diagonal, we consider two different access models. In the first case, given a general invertible, 1-sparse matrix $A \in \mathbb{C}^{N\times N}$, it can be written as $A = \Pi D$, where Π is a permutation matrix, and D is a diagonal matrix. We assume that we have direct access to the permutation Π. Then A is invertible if and only if D is invertible. Given the availability of an $(1, m'_\Pi, 0)$ block encoding of the unitary matrix Π^{-1} denoted by U'_Π, as well as an $(\alpha'_D, m'_D, \epsilon)$ block encoding of D^{-1} denoted by U'_D, we obtain an $(\alpha'_D, m'_D + m'_\Pi, \epsilon)$ block encoding of $A^{-1} = D^{-1}\Pi^{-1}$ via multiplication of block-encoded matrices [7]. The whole circuit takes three oracle queries in total.

In the second case, we only assume we have query access to the column of the single nonzero element in each row (since

the matrix is 1-sparse), as well as to the value of the each element. The details for constructing the fast inversion in this case are given in Appendix C.

B. Fast inversion of normal matrices

If $A \in \mathbb{C}^{N\times N}$ is a normal matrix, i.e., $AA^\dagger = A^\dagger A$, then A can be unitarily diagonalized as $A = VDV^\dagger$, where $V \in \mathbb{C}^{N\times N}$ is a unitary matrix and $D \in \mathbb{C}^{N\times N}$ is a diagonal matrix. Therefore, A is invertible if and only if D is invertible. Using the fast-inversion routine, we obtain an efficient block encoding of A^{-1} as

$$U'_A = (V \otimes I_{l+1})U'_D(V^\dagger \otimes I_{l+1}). \tag{10}$$

Therefore, we have the following proposition:

Proposition 2 (Fast inversion for normal matrices). Suppose the eigenvalues of a normal matrix $A = VDV^\dagger$, where V is unitary and D is diagonal, can be accessed through the oracle O_D given in Eq. (7), and V can be efficiently implemented in a quantum circuit. Also, let $\alpha'_D \geqslant 1/\min_i\{|D_{ii}|\}$, then we can construct an $(\alpha'_D, m'_D, 0)$ block encoding of A^{-1}, using O_D, $O_D^\dagger$, V, and $V^\dagger$ each exactly once. Here $m'_D = O(l + \text{polylog}(N))$ if O_D uses $O(l + \text{polylog}(N))$ ancilla qubits.

Let us now consider another example of solving a linear system via fast inversion. Consider the following d-dimensional elliptic equation with periodic boundary conditions:

$$-\Delta u(\mathbf{r}) + u(\mathbf{r}) = b(\mathbf{r}), \quad \mathbf{r} \in \Omega = [0, 1]^d. \tag{11}$$

Using a plane-wave basis set, we may expand u, b as

$$u(\mathbf{r}) = \sum_{\mathbf{G}\in\mathbb{G}} \hat{u}(\mathbf{G}) \exp(i\mathbf{G}\cdot\mathbf{r}),$$
$$b(\mathbf{r}) = \sum_{\mathbf{G}\in\mathbb{G}} \hat{b}(\mathbf{G}) \exp(i\mathbf{G}\cdot\mathbf{r}),$$

where the plane-wave index set is

$$\mathbb{G} = \left\{ \mathbf{G} = 2\pi(g_1, \ldots, g_d) \;\middle|\; g_i \in \mathbb{Z}, i = 1, \ldots, d \right\}.$$

The solution can then be readily written in the Fourier space as

$$\hat{u}(\mathbf{G}) = \frac{1}{|\mathbf{G}|^2 + 1}\hat{b}(\mathbf{G}), \quad \mathbf{G} \in \mathbb{G}.$$

We now use a finite number of N plane waves to approximate the solution to Eq. (11). For simplicity we assume $N = 2^n = 2^{d\mathfrak{n}}$, so that there are $h^{-1} := N^{1/d} = 2^{\mathfrak{n}}$ plane waves per dimension. We further assume here h can be viewed as an effective mesh size. Then, the plane-wave indices are restricted to

$$\mathbb{G}_h = \left\{ \mathbf{G} = 2\pi(g_1, \ldots, g_d) \mid -\frac{1}{2h} \leqslant g_i < \frac{1}{2h}, \right.$$
$$\left. g_i \in \mathbb{Z},\ i = 1, \ldots, d \right\}, \tag{12}$$

with the cardinality $|\mathbb{G}_h| = N$, and the resulting discretized QLSP can be written as $A|u\rangle \propto |b\rangle$. We may write $A = VDV^\dagger$, where V is the d-dimensional quantum Fourier transform (QFT) [48], and the diagonal entries of D are known and labeled by $\mathbf{G}$ as

$$D(\mathbf{G}) = |\mathbf{G}|^2 + 1, \quad \mathbf{G} \in \mathbb{G}_h.$$

Hence, the largest singular value of A is

$$\|A\| = \sigma_{\max} = d\frac{(2\pi)^2}{(2h)^2} + 1 = \frac{d\pi^2}{h^2} + 1,$$

which grows as h^{-2} as the number of plane waves increases. The smallest singular value of A is $1/\|A^{-1}\| = \sigma_{\min} = 1$. Therefore, the condition number $\kappa(A) = O(dh^{-2})$. A block encoding of D^{-1} can be explicitly constructed using classical arithmetics, and therefore we may fast invert the matrix A.

Let us consider $d = 1$ first, where $n = \mathfrak{n}$ and V is the standard QFT for $\mathfrak{n}$ qubits, denoted by $\mathrm{F}_\mathfrak{n}$. The implementation of $\mathrm{F}_\mathfrak{n}$ costs $O(\mathfrak{n}^2)$ gates. In the d-dimensional setting, V can be constructed by d copies of $\mathrm{F}_\mathfrak{n}$ as

$$V = \mathrm{F}_\mathfrak{n} \otimes \cdots \otimes \mathrm{F}_\mathfrak{n}.$$

So the total cost is $\widetilde{O}(\mathfrak{n}d) = \widetilde{O}(n/d)$. Therefore, the circuit depth for block encoding A^{-1} is independent of the condition number $\kappa(A)$.

To consider the query complexity, we first note that the norm of the solution

$$\int |u(\mathbf{r})|^2 d\mathbf{r} = \sum_{\mathbf{G}\in\mathbb{G}} |\hat{u}(\mathbf{G})|^2 = \sum_{\mathbf{G}\in\mathbb{G}} \frac{1}{(|\mathbf{G}|^2 + 1)^2}|\hat{b}(\mathbf{G})|^2 \tag{13}$$

is well defined as long as the Fourier coefficients of the right-hand side $\hat{b}(\mathbf{G})$ decay rapidly enough as $|\mathbf{G}| \to \infty$ (e.g., when $b(\mathbf{r})$ is a smooth function). Therefore, with a finite truncation

$$\sum_{\mathbf{G}\in\mathbb{G}_h} \frac{1}{(|\mathbf{G}|^2 + 1)^2}|\hat{b}(\mathbf{G})|^2 = \Theta(1),$$

and the quantity

$$\xi = \|A^{-1}|b\rangle\| = \Theta(1).$$

This is asymptotically the best scenario for solving QLSP as discussed in Sec. II B and Appendix B. Combining the results of our bound on the complexity of the linear systems problem (given in Appendix B as Theorem 7) and Proposition 2, we have the following proposition for the cost of solving the elliptic equation (11).

Proposition 3. In order to solve the d-dimensional elliptic equation (11) with a smooth right-hand side $b(\mathbf{r})$ on the d-dimensional torus with precision ϵ and success probability larger than $\frac{1}{2}$, using a plane-wave discretization (12) with grid size h along each direction, the circuit depth of the quantum singular value transformation is $O(dh^{-2}\ln(1/\epsilon))$. The total number of queries to $U_A, U_A^\dagger$ is $O(dh^{-2}\ln(1/\epsilon))$, and the number of queries to U_b is $O(1)$. The circuit depth, number of queries to $O_D, O_D^\dagger, V, V^\dagger, U_b$ are all $O(1)$ using fast inversion.

In the example above, A is a Hermitian matrix. If we replace A by a normal matrix $A' = A - zI$ with $z \in \mathbb{C}$ so that $|z| \ll \|A\|$ and A' is invertible, the conclusion still holds. This is the case in the contour-integral formulation of computing matrix functions in Sec. V, and in the computation of Green's functions in Sec. IV.

Remark 4. In the context of solving the d-dimensional Poisson equation, the fast-inversion method is different from the method in [4] using the HHL algorithm, which employs the Hamiltonian simulation of the form e^{-iAt}. This requires $\Omega(\ln(1/\epsilon))$ ancilla qubits to store the eigenvalues, and if the Hamiltonian simulation is implemented directly, the circuit depth is $\Omega(\|A\|t)$. On the other hand, Eq. (10) is based on the direct access of oracles O_D and U'_D, and the QFT part is decoupled from the inversion part. Neither the circuit depth nor the number of ancilla qubits needed depends on $\|A\|$ or the accuracy ϵ.

Remark 5 (Complexity lower bound with respect to ξ). As can be seen in the above discussion, when A is fast invertible, solving $A|x\rangle \propto |b\rangle$ can still have a $1/\xi$ dependence, where $\xi = \|A^{-1}|b\rangle\|$, and we assume for simplicity $\|A^{-1}\| = 1$. This dependence is in fact the best we can get. Consider the following simple example constructed from the unstructured search problem with n bits and $N = 2^n$: let U_w be the oracle for the unstructured search problem marking the target element w through

$$U_w|s\rangle = \begin{cases} |s\rangle, & s \neq w \\ -|s\rangle, & s = w. \end{cases}$$

Now we let

$$A = \frac{\sqrt{N}-1}{2}U_w + \frac{\sqrt{N}+1}{2}I,$$

and this is a diagonal matrix whose diagonal entries are efficiently computable, and therefore fast invertible. Solving the QLSP $A|x\rangle \propto |u\rangle$, where $|u\rangle$ is the uniform superposition of all n-bit strings, results in a solution

$$|x\rangle = \frac{\sqrt{N}}{\sqrt{2N-1}}|w\rangle + \frac{1}{\sqrt{2N-1}}\sum_{s\neq w}|s\rangle$$

with

$$\xi = \|A^{-1}|u\rangle\| = \left\| \frac{1}{\sqrt{N}}|w\rangle + \frac{1}{N}\sum_{s\neq w}|s\rangle \right\| = \Theta(N^{-1/2}).$$

If the QLSP with this A can be solved with $o(1/\xi)$ queries to A, then it means we can obtain $|x\rangle$ with $o(\sqrt{N})$ queries to U_w. Measuring all qubits with the state $|x\rangle$ yields w with probability around $\frac{1}{2}$. Therefore, we would be able to solve the unstructured search problem with query complexity $o(\sqrt{N})$, which is impossible. We therefore think the name "fast inversion" for our method to deal with this kind of QLSP is appropriate because it cannot be asymptotically improved without further assumptions.

III. PRECONDITIONED QLSP SOLVER

We consider the following linear system (1), with a large condition number $\kappa(A+B)$, where A and B are n-qubit matrices. We are primarily interested in the following scenario: we assume $A+B$ is rescaled so that $\|(A+B)^{-1}\| = \Theta(1)$ and $\|A\| \gg \|B\|$, $\|(A+B)^{-1}\|$, $\|A^{-1}\|$ (if needed we may replace A and B with $A - zI$ and $B + zI$ for some $z \in \mathbb{C}$). The condition number $\kappa := \kappa(A+B) = \Theta(\|A\|)$. The linear system is therefore ill conditioned mainly as a result of the large $\|A\|$.

We make the following assumptions regarding the query access in this problem. We assume we have U'_A, an $(\alpha'_A, m'_A, 0)$ block encoding of A^{-1} prepared by the fast-inversion procedure, and U_B, an $(\alpha_B, m_B, 0)$ block encoding of B. For simplicity of presentation we assume these given block encodings are error free (see Remark 1). The right-hand side b is accessed through a quantum circuit U_b, i.e., $|b\rangle = U_b|0^n\rangle$.

In the algorithm we need multiple ancilla registers, and will refer to the register in which $|b\rangle$ is prepared and $|x\rangle$ is produced as the main register (also called the system register).

A. Preconditioning the linear system

It is possible to reduce the condition number of the linear system by considering the following equivalent formulation:

$$(I + A^{-1}B)|x\rangle \propto A^{-1}|b\rangle. \tag{14}$$

Lemma 1 explains why this linear system might have a much smaller condition number than the linear system (1).

Lemma 1. Define $W = I + A^{-1}B$, then the smallest singular value $\sigma_{\min}$ and largest singular value $\sigma_{\max}$ of W satisfy

$$1/\sigma_{\min} \leqslant 1 + \|(A+B)^{-1}\|\,\|B\| =: C_{AB},$$
$$\sigma_{\max} \leqslant 1 + \|A^{-1}\|\,\|B\| =: C'_{AB}.$$

Hence, the condition number of W can be upper bounded as

$$\kappa(W) \leqslant [1 + \|(A+B)^{-1}\|\,\|B\|](1 + \|A^{-1}\|\,\|B\|) = C_{AB}C'_{AB}. \tag{15}$$

Proof. Let $W|x\rangle = |y\rangle$. Then we have

$$(A+B)|x\rangle = A|y\rangle,$$

therefore,

$$(A+B)(|x\rangle - |y\rangle) = -B|y\rangle,$$
$$A(|x\rangle - |y\rangle) = -B|x\rangle,$$

and these two equalities lead to

$$\|\,|x\rangle\| \leqslant \|\,|y\rangle\| + \||x\rangle - |y\rangle\| \leqslant [1 + \|(A+B)^{-1}\|\,\|B\|]\|\,|y\rangle\|,$$
$$\|\,|y\rangle\| \leqslant \|\,|x\rangle\| + \||x\rangle - |y\rangle\| \leqslant (1 + \|A^{-1}\|\,\|B\|)\|\,|x\rangle\|.$$

These two inequalities then give a lower bound for the smallest singular value, and an upper bound for the largest singular value, as stated in the lemma. ■

Remark 6. The upper bound of $\kappa(W)$ does not depend on $\|A\|$ which we assume to be the main reason why the linear system (1) is ill conditioned. For a given pair of A and B we can always rescale A and B, and possibly shifting by a multiple of identity, i.e., consider instead $A - \mu I$ and $B + \mu I$, so that the smallest singular values of $A+B$ and A are $\Omega(1)$. Equation (15) then gives us a bound for the condition number of W that is independent of $\|A\|$. When $\|B\|^2 \ll A$, we have $\kappa(W) \ll \kappa(A+B)$.

Our bound of $1/\sigma_{\min}$ scales linearly with respect to $\|B\|$. So $\kappa(W)$ may scale quadratically with respect to $\|B\|$, leading to an undesirable polynomial dependence on the block-encoding subnormalization factor of B in later applications. However, such estimate can be overly pessimistic in practice. In Fig. 2 we plot the smallest singular value of the matrix W corresponding to discretizing a differential operator

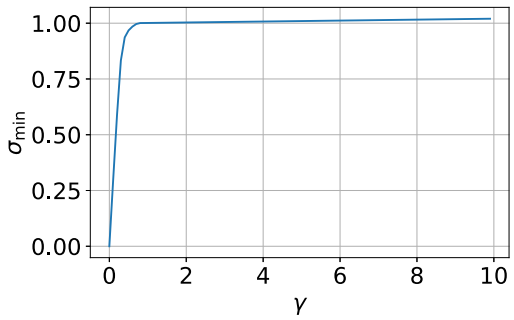


FIG. 2. The smallest singular value of $W = I + A^{-1}B$ where A and B come from discretizing the differential operator $-\Delta + \gamma[3 + \cos(5x)]$ with periodic boundary condition in 1D. $A = -\Delta_h + I$ where Δ_h is the discrete Laplacian operator, and $B = \gamma V - I$ where V is the diagonal matrix whose diagonal elements contain the potential.

$-\Delta + \gamma[3 + \cos(5x)]$ with periodic boundary condition in 1D. In this example $\|A\|$ dominates because the Laplacian operator is unbounded in the L^2 norm. We can see that instead of going to 0 as γ increases, $\sigma_{\min}$ actually increases. Therefore, $1/\sigma_{\min}$ in this example can be bounded by a constant that does not depend on $\|B\|$ for γ bounded away from 0.

In Secs. IV A and IV B, we will see this procedure can be applied to many linear systems of practical interest. Next we consider how to construct a quantum circuit to solve the preconditioned linear system (14).

B. Quantum circuit construction

We want to construct a quantum circuit to block encode W^{-1} and thereby solve the preconditioned linear system. This is done by using QSVT (see Appendix B). In the following, we first construct a block encoding for W. We then use QSVT to obtain a block encoding of W^{-1} to solve the linear system.

For the first step, since we can first apply the multiplication of block encodings [45, Lemma 30] to obtain an $(\alpha'_A\alpha_B, m'_A + m_B, 0)$ block encoding of $A^{-1}B$, then apply the linear combination of block encodings [7, Lemma 29] to obtain an $(\alpha'_A\alpha_B + 1, m'_A + m_B + 1, 0)$ block encoding of $W = I + A^{-1}B$, which we denote by U_W.

Following [45, Corollary 69], we may construct an odd polynomial $P(x)$ of degree $O(\frac{1}{\delta}\ln(\frac{1}{\epsilon'}))$ that satisfies

$$\left|P(x) - \frac{3\delta}{4x}\right| \leqslant \epsilon', \quad x \in [-1, -\delta] \cup [\delta, 1].$$

Suppose the smallest singular value of W is lower bounded by $\widetilde{\sigma}_{\min}$. By Lemma 1 we can choose $\widetilde{\sigma}_{\min} = 1/C_{AB}$. We will apply the polynomial to implement $[P^\diamond(W/(\alpha'_A\alpha_B + 1))]^\dagger$, where $P^\diamond(\cdot)$ denotes the generalized matrix function as defined in Definition 4. We intend to use this to approximate $(W/(\alpha'_A\alpha_B + 1))^{-1}$. Therefore, we need to ensure the singular values of $W/(\alpha'_A\alpha_B + 1)$ lie in the interval $[\delta, 1]$. Recall that a lower bound of the smallest singular value of W is $\widetilde{\sigma}_{\min}$. For this purpose we choose

$$\delta = \widetilde{\sigma}_{\min}/(\alpha'_A\alpha_B + 1).$$

With this odd polynomial $P(x)$, we use QSVT [7, Corollary 11] to construct a circuit to block encode the matrix obtained by applying $P(x)$ to singular values of $W/(\alpha'_A\alpha_B + 1)$, using only one extra ancilla qubit (see Appendix B and Fig. 4 for the circuit). This is a $(1, m'_A + m_B + 2, 0)$ block encoding of $P^\diamond(W/(\alpha'_A\alpha_B + 1))$. We denote the Hermitian conjugate [see Eq. (B5) in Appendix B] of the block encoding constructed in this way by U'_W. U'_W is therefore a $(\frac{4}{3\widetilde{\sigma}_{\min}}, m'_A + m_B + 2, \epsilon'')$ block encoding of W^{-1}, where

$$\epsilon'' = \frac{4\epsilon'}{3\widetilde{\sigma}_{\min}}.$$

In other words,

$$\left\| \frac{4}{3\widetilde{\sigma}_{\min}}(\langle 0^l| \otimes I)U'_W(|0^l\rangle|y\rangle) - W^{-1}|y\rangle \right\| \leqslant \epsilon'',$$

where the first register contains $l = m'_A + m_B + 2$ ancilla qubits, and the second register is the main register.

From the above analysis we have obtained a block encoding of W^{-1}. Note that $(A + B)^{-1} = W^{-1}A^{-1}$ is a product of two block-encoded matrices. Therefore, by the multiplication of block encodings [7, Lemma 30] we have a $(\frac{4\alpha'_A}{3\widetilde{\sigma}_{\min}}, 2m'_A + m_B + 3, \alpha'_A\epsilon'')$ block encoding of $(A + B)^{-1}$. This only uses U'_A one extra time. We want the block-encoding error to be $\alpha'_A\epsilon'' = \delta'$, then we need to choose

$$\epsilon' = \frac{3\delta'\widetilde{\sigma}_{\min}}{4\alpha'_A} \tag{16}$$

and the polynomial degree can then be expressed with respect to δ' as

$$O\left(\frac{1}{\delta}\ln\left(\frac{1}{\epsilon'}\right)\right) = O\left(\frac{\alpha'_A\alpha_B}{\widetilde{\sigma}_{\min}}\ln\left(\frac{\alpha'_A}{\delta'\widetilde{\sigma}_{\min}}\right)\right).$$

The cost of applying QSVT scales linearly with the polynomial degree. We can then summarize the result in the following theorem:

Theorem 1 (Block encoding of preconditioned matrix inverse). Let U'_A be an $(\alpha'_A, m'_A, 0)$ block encoding of A^{-1} implemented via fast inversion, U_B be an $(\alpha_B, m_B, 0)$ block encoding of B. Let $\widetilde{\sigma}_{\min}$ be a lower bound for the smallest singular value of $I + A^{-1}B$, which can be chosen to be $1/[1 + \|(A + B)^{-1}\|\|B\|]$ as discussed in Lemma 1. Then for any $\delta' > 0$ there exists a $(\frac{4\alpha'_A}{3\widetilde{\sigma}_{\min}}, 2m'_A + m_B + 3, \delta')$ block encoding of $(A + B)^{-1}$ using $O(\frac{\alpha'_A\alpha_B}{\widetilde{\sigma}_{\min}}\ln(\frac{\alpha'_A}{\delta'\widetilde{\sigma}_{\min}}))$ applications of U'_A, U_B, their controlled versions, their inverses, and other primitive gates.

There are many parameters involved in the above discussion which can be confusing to readers. Here we briefly summarize their relations. The complexity depends directly on two parameters: δ, which is how far the singular values of $W/(\alpha'_A\alpha_B + 1)$ are bounded away from 0, and ϵ', which is the error of polynomial approximation. In the block encoding of W^{-1}, the polynomial approximation error is amplified into the block-encoding error ϵ'', which is then amplified into the final block-encoding error δ' through the multiplication of two block-encoded matrices. We assume δ' is chosen *a priori* and therefore it requires us to choose ϵ' according to Eq. (16).

Next, we want to solve the linear system $|x\rangle \propto (A+B)^{-1}|b\rangle$ where $|x\rangle$ is the normalized solution to the linear system. We denote the block encoding of $(A+B)^{-1}$ in the above theorem by $\mathcal{U}$. We denote

$$|w\rangle = \frac{4\alpha'_A}{3\widetilde{\sigma}_{\min}}(\langle 0^{l'}| \otimes I)\mathcal{U}(|0^{l'}\rangle \otimes |b\rangle),$$

where $l' = 2m'_A + m_B + 3$. Therefore, $|w\rangle/\||w\rangle\|$ is the output state after applying the block encoding. Also, we denote $|y\rangle = (A+B)^{-1}|b\rangle$, $\zeta = \||w\rangle\|$, and $\xi = \||y\rangle\|$. Thus, the normalized solution $|x\rangle = |y\rangle/\||y\rangle\|$. Then we have

$$\||w\rangle - |y\rangle\| \leqslant \delta', \quad |\zeta - \xi| \leqslant \delta'.$$

This leads to

$$\left\| \frac{|w\rangle}{\zeta} - \frac{|y\rangle}{\xi} \right\| \leqslant \frac{\zeta\||w\rangle - |y\rangle\| + |\zeta - \xi|\||w\rangle\|}{\zeta\xi} \leqslant \frac{2\delta'}{\xi}.$$

Therefore, in order to make sure the output normalized quantum state $|w\rangle/\||w\rangle\|$ is ϵ close to $|x\rangle$ in terms of 2-norm distance, we need $\delta' = \xi\epsilon/2$. The success probability is

$$\|(\langle 0^{l'}| \otimes I)\mathcal{U}(|0^{l'}\rangle \otimes |b\rangle)\|^2 = \frac{9\||w\rangle\|^2\widetilde{\sigma}^2_{\min}}{16{\alpha'_A}^2} \geqslant \frac{9\xi^2(1-\epsilon/2)^2\widetilde{\sigma}^2_{\min}}{16{\alpha'_A}^2}.$$

We can boost the success probability to be greater than $\frac{1}{2}$ by amplitude amplification, using $O(\alpha'_A/\xi\widetilde{\sigma}_{\min})$ repetitions. Therefore, we have the following corollary:

Corollary 1 (Preconditioned linear system solver). Under the same assumptions as Theorem 1, for the QLSP (1), an ϵ-close solution vector can be obtained with $O(\frac{{\alpha'_A}^2\alpha_B}{\xi\widetilde{\sigma}^2_{\min}}\ln(\frac{\alpha'_A}{\widetilde{\sigma}_{\min}\xi\epsilon}))$ applications of U'_A, U_B, their controlled versions, their inverses, and other primitive gates, in addition to $O(\frac{\alpha'_A}{\widetilde{\sigma}_{\min}\xi})$ applications of U_b and its inverse, where $\xi = \|(A+B)^{-1}|b\rangle\|$. As in Theorem 1, $\widetilde{\sigma}_{\min}$ can be chosen to be $1/[1 + \|(A+B)^{-1}\|\,\|B\|]$.

Below we compare the dependence on the condition number of our preconditioning method against the dependence of directly using QSVT. Let us consider the scenario we proposed at the beginning of Sec. III. We assume a rescaling is applied to $A+B$ so that $\|(A+B)^{-1}\| = \Theta(1)$, $\|A\| \to \infty$, and $\|A^{-1}\|, \|B\| = O(1)$. As discussed before if $\|A^{-1}\|$ is large we can always replace it with $A - zI$ for some $z \in \mathbb{C}$ that is away from the spectrum of A. Furthermore, we assume α'_A and α_B are not much larger than $\|A^{-1}\|$ and $\|B\|$, i.e., $\alpha'_A, \alpha_B = O(1)$. We also assume $\epsilon = \Omega(1)$ so that we do not need to consider the dependence on ϵ. This is because both methods have a logarithmic dependence on $1/\epsilon$ and are therefore similar in this aspect.

Under these assumptions we have

$$\kappa(A+B) = \|A+B\|\|(A+B)^{-1}\| = \Theta(\|A+B\|) = \Theta(\|A\|). \tag{17}$$

The smallest singular value of $W = I + A^{-1}B$ is lower bounded by

$$\widetilde{\sigma}_{\min} = 1/[1 + \|(A+B)^{-1}\|\,\|B\|] = \Omega(1).$$

From Corollary 1, we can see that the number of queries to all oracles become $O(\frac{1}{\xi}\ln(\frac{1}{\xi}))$. This is no longer directly dependent on $\kappa(A+B)$, though such dependence can exist indirectly through the dependence of ξ on $\kappa(A+B)$. We consider the following two cases:

(1) In the worst case, the following inequality becomes an equality: $\xi = \|(A+B)^{-1}|b\rangle\| \geqslant \|A+B\|^{-1} = \Omega[1/\kappa(A+B)]$ by Eq. (17). Therefore, we have $O(\kappa(A+B)\ln[\kappa(A+B)])$ query complexity for U'_A, U_B, and $O(\kappa(A+B))$ query complexity for U_b.

(2) In the best case, $|b\rangle$ has $\Omega(1)$ overlap with the left singular vectors of $A+B$ corresponding to small singular values, and therefore ξ can be as large as $\Omega(1)$, giving us a query complexity of $O(1)$ for all oracles. For a concrete example of this scenario, see Proposition 3.

In both cases we can compare with direct application of QSVT as discussed in Appendix B. The worst and best scenarios are discussed for QSVT without preconditioning in Appendix B. In the worst case, under the assumption that the block encoding of $A+B$, denoted by U_{A+B}, entails a subnormalization factor $\|A+B\|$, direct application of QSVT will need to query U_{A+B} and its inverse $O(\kappa(A+B)^2\ln[\kappa(A+B)])$ times, and U_b and its inverse $O(\kappa(A+B))$ times. The preconditioning method results in a quadratic improvement. In the best case, i.e., $|b\rangle$ has $\Omega(1)$ overlap with the left singular vectors of $A+B$ corresponding to small singular values, under the same assumption, the number of queries to U_{A+B} is $O(\kappa(A+B))$ and the number of queries to U_b is $O(1)$. This improvement comes from the fact that the large overlap makes the success probability of applying the QSVT bounded away from zero by a constant. The improvement of using preconditioning in this case is more significant, as we can dispense with this linear dependence on the condition number altogether.

We remark that the speedup in our method does not come from a reduced effective condition number. Consider the following simple example: let $|u_{\min}\rangle$ and $|u_{\max}\rangle$ be the left-singular vectors of $A+B$ associated with the smallest and largest singular values, respectively. Then for the QLSP $(A+B)|x\rangle \propto |b\rangle$ where $|b\rangle = \frac{1}{\sqrt{2}}(|u_{\min}\rangle + |u_{\max}\rangle)$, we have $\xi = \Theta(1)$, and in the scenario discussed above solving the QLSP only requires $O(1)$ queries to all oracles. The effective condition number of this problem is, however, the same as the condition number of $A+B$, which is $\Theta(\|A\|)$.

Remark 7. The above procedure assumes we know the constants such as $C_{AB} = 1 + \|(A+B)\|^{-1}\|B\|$ and ξ. The algorithm still works if upper bounds to these constants are known. In Theorem 1 and Corollary 1, a superlinear dependence on the block-encoding factor α_B (or alternatively the matrix norm $\|B\|$) will arise if we use the bound in Lemma 1, letting $\widetilde{\sigma}_{\min} = 1/C_{AB}$. However, according to the discussion in Remark 6, it is possible that $\widetilde{\sigma}_{\min}$ can be independent of α_B. In this case, the preconditioned linear system solver can scale linearly with respect to α_B.

IV. EVALUATING GREEN'S FUNCTIONS OF QUANTUM MANY-BODY SYSTEMS

We first give a short introduction of the representation of fermionic systems. We consider here a second-quantized representation wherein each state is referred to as an orbital. The Pauli exclusion principle forbids two electrons from being

in the same spin state and spatial state simultaneously. Since spin for an electron can either be up or down, there are four possible occupation states for each orbital. This means that two qubits are needed to represent a given configuration of an orbital. For example, the qubit states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are taken to represent an orbital containing no electrons, no spin down and one spin up, one spin down and no spin up, and one spin up and down electron, respectively.

Since an orbital is naturally expressed as a pair of qubits in quantum computing, it is natural to further divide an orbital into two spin orbitals which correspond to both the quantum states for both the spin and spatial degrees of freedom. In this notation, we can describe the occupation and dynamics of a set of spin orbitals using creation and annihilation operators such that a spin orbital corresponding to spin state σ and spatial orbital ν is given by the state $\hat{a}^\dagger_{\nu,\sigma}|0\rangle_{\nu,\sigma} = |1\rangle_{\nu,\sigma}$ where $\hat{a}^\dagger_{\nu,\sigma}$ is a fermionic (anticommuting) creation operator acting on the spin orbital. Similarly, $\hat{a}^\dagger_{\nu,\sigma}|1\rangle_{\nu,\sigma} = 0$. This leads us to the number operator $\hat{n}_{\nu,\sigma} = \hat{a}^\dagger_{\nu,\sigma}\hat{a}_{\nu,\sigma}$, which has the property that $\hat{n}_{\nu,\sigma}\hat{a}^\dagger_{\nu,\sigma}|0\rangle_{\nu,\sigma} = \hat{a}^\dagger_{\nu,\sigma}|0\rangle_{\nu,\sigma}$ and $\hat{n}_{\nu,\sigma}|0\rangle_{\nu,\sigma} = 0$. Thus, this operator is often called a number operator because it counts the number of electrons in a spin orbital.

We may also use a single index i to represent the multi-index (ν, σ). Then the creation and annihilation operators $\hat{a}^\dagger_i, \hat{a}_i$ can be expressed using the Pauli operator via, e.g., the Jordan-Wigner transform [32] as

$$\hat{a}_i = Z^{\otimes(i-1)} \otimes \tfrac{1}{2}(X + iY) \otimes I^{\otimes(N-i)},$$
$$\hat{a}^\dagger_i = Z^{\otimes(i-1)} \otimes \tfrac{1}{2}(X - iY) \otimes I^{\otimes(N-i)}. \tag{18}$$

Correspondingly, the number operator can be represented as

$$\hat{n}_i = \tfrac{1}{2}(I - Z_i). \tag{19}$$

Here X, Y, Z, I are single-qubit Pauli matrices. Note that Eqs. (18) and (19) naturally provide a (1,1,0) block encoding of $\hat{a}_i, \hat{a}^\dagger_i, \hat{n}_i$.

As a practical application of the preconditioned linear system solver, we consider the evaluation of the single-particle Green's function in quantum many-body systems. The fermionic Hamiltonian (in the spin-orbital formulation [32]) can be naturally separated into the sum of two terms as

$$\hat{H} = \hat{H}_0 + \hat{H}_1.$$

Here $\hat{H}_0, \hat{H}_1$ are the noninteracting part and interacting part of the Hamiltonian, respectively:

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij}\hat{a}^\dagger_i\hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{ijkl}\hat{a}^\dagger_i\hat{a}^\dagger_j\hat{a}_l\hat{a}_k. \tag{20}$$

In this section, N is the number of spin orbitals used to discretize the continuous Hamiltonian, and the dimension of the Hamiltonian matrix $\hat{H}$ is 2^N.

We denote by $|\Psi_0\rangle$ the ground state of $\hat{H}$ with N_e electrons ($N_e \leqslant 2N$), and E_0 is the corresponding ground-state energy. We assume that the ground state $|\Psi_0\rangle$ can be prepared by an oracle with error ς and success probability at least p, and E_0 is known to some error ς'. We will provide examples of the realization of $\hat{H}_0, \hat{H}_1$ in Sec. IV C.

Remark 8 (Complexity of solving the ground-state problem). The above assumptions are, in general, unlikely to be satisfied for generic fermionic systems. This assumption is of course very strong because if it were true in general then QMA $\subseteq$ BQP, which is widely believed to be false. Even the problem of deciding whether the ground-state energy is above or below a threshold (within a fixed promise gap) is known to be in QMA hard [49–52]. Nonetheless, it is reasonable to make these assumptions for systems where an ansatz can be constructed that has polynomial overlap with the ground state, and where the spectral gap of the Hamiltonian can be bounded from below. These are the assumptions made in, e.g.. [53,54], and is believed to occur for a wide range of realistic systems in physics and chemistry.

A. Single-particle Green's function

The single-particle Green's function is a matrix-valued function (formally mapping $\mathbb{C} \mapsto \mathbb{C}^{N\times N}$ matrix) that we denote $G(z)$. Here the input z can often be interpreted to be an energy shift and $G(z)$ is defined provided $E_0 + z$ is not an eigenvalue of H. Also, note that the dimension of the underlying Hilbert space for the problem is 2^N, the matrix is relatively small compared to the dimension of the Hamiltonian. We first define the advanced and retarded Green's function [denoted by $G^{(+)}(z)$ and $G^{(-)}(z)$, respectively] as

$$G^{(+)}_{ij}(z) := \langle\Psi_0|\hat{a}_i(z - [\hat{H} - E_0])^{-1}\hat{a}^\dagger_j|\Psi_0\rangle,$$
$$G^{(-)}_{ij}(z) := \langle\Psi_0|\hat{a}^\dagger_j(z + [\hat{H} - E_0])^{-1}\hat{a}_i|\Psi_0\rangle. \tag{21}$$

Then the time-ordered single-particle Green's function, or Green's function for short, is the sum of the two components

$$G(z) = G^{(+)}(z) + G^{(-)}(z).$$

We assume $|\operatorname{Im}(z)| \geqslant \eta$. The value of η is often referred to as the broadening parameter, and determines the resolution of Green's functions along the energy spectrum.

Below we demonstrate a procedure that allows us to directly compute $G^{(\pm)}_{ij}(z)$. Now suppose we have an $(\alpha^{(+)}, m^{(+)}, \epsilon^{(+)})$ block encoding of the matrix inverse $(z - [\hat{H} - E_0])^{-1}$ denoted by $U^{(+)}$, then using the Jordan-Wigner transformation (18) and the block encoding of product of matrices [7, Lemma 30], we can construct an $(\alpha^{(+)}, m^{(+)} + 2, \epsilon^{(+)})$ block encoding of $\hat{a}_i(z - [\hat{H} - E_0])^{-1}\hat{a}^\dagger_j$, which we denote by $\widetilde{U}^{(+)}$. Then, the Hadamard test for nonunitary matrices in Appendix D tells us how to estimate $G^{(+)}(z)$. The same procedure can be applied to obtain $G^{(-)}(z)$.

We remark that if we are only interested in the imaginary part of the Green's function [or, more accurately, the anti-Hermitian part of the Green's function defined as $\Gamma^{(\pm)} := \frac{1}{2i}(G^{(\pm)} - G^{(\pm)\dagger})$], then we can directly use amplitude estimation without using the Hadamard test, which saves us one control qubit. $\Gamma^{(\pm)}$ is related to the spectral functions of the many-body system. The details of computing $\Gamma^{(\pm)}$ are discussed in Appendix E.

B. Preconditioned Green's-function solver

As can be seen from the above discussion, matrix inversion is a crucial part of evaluating Green's function. In this section, we use the preconditioning technique developed in Sec. III to efficiently perform the matrix inversion. Unlike in the QLSP

setting in Corollary 1, the performance of Green's-function solvers does not depend on the amplitude $\xi = \|A^{-1}|b\rangle\|$.

It now only remains to determine the block encoding of $(z - \hat{H} + E_0)^{-1}$. Since $|\operatorname{Im} z| \geqslant \eta$, the smallest singular value of $z - \hat{H} + E_0$ is bounded from below by η, and $\|(z - \hat{H} + E_0)^{-1}\| \leqslant \eta^{-1}$. If $\|\hat{H}_0\|$ and $\|\hat{H}_1\|$ are comparable, it is natural to consider constructing this block encoding using existing QLSP solvers such as HHL or the ones based on LCU or QSVT. The complexities of these direct approaches are in Table I, and for completeness the analysis is in Appendix F. Direct block encoding the matrix inverse $(z - \hat{H} + E_0)^{-1}$ using LCU or QSVT results in a linear dependence on $\|\hat{H}\|$ in the query complexity (here we assume the block-encoding factor of $z - \hat{H} + E_0$ is comparable to $\|\hat{H}\|$), as shown in Table I. However, in certain physical settings, we may have $\|\hat{H}_0\| \gg \|\hat{H}_1\|$ or $\|\hat{H}_1\| \gg \|\hat{H}_0\|$ (see Sec. IV C). Then we may reduce the complexity through preconditioned linear system solvers in Theorem 1. As shown in Table I, our method enables us to replace the dependence on $\|\hat{H}\|$ with a dependence on the smaller one between $\|\hat{H}_0\|$ and $\|\hat{H}_1\|$.

According to Remark 8, we assume the ground energy is known to a precision ς', and the ground state can be prepared to within trace-distance error ς with probability at least p by some circuit U_Ψ. In the analysis below we first ignore the error of the ground energy for simplicity, but add back its contribution at the end.

Without loss of generality, we repartition the Hamiltonian as $\hat{H} = \hat{A} + \hat{B}$, where $\|\hat{A}\| \gg \|\hat{B}\|$, and $\hat{A}$ can be efficiently unitarily diagonalized as in Proposition 2. In order to use the preconditioning technique in Sec. III, we first split $z - \hat{H} + E_0$ into the sum of $z + i - \hat{A} + E_0$ and $\hat{B} - i$ for $\operatorname{Im}(z) > 0$, or $z - i - \hat{A} + E_0$ and $\hat{B} + i$ for $\operatorname{Im}(z) < 0$. An extra shift $\pm i$ is introduced to so that $\|(z \pm i - \hat{A} + E_0)^{-1}\| \leqslant 1$, and $z \pm i - \hat{A} + E_0$ is still a normal matrix and can be fast inverted.

For simplicity we first assume $\operatorname{Im}(z) > 0$. We can construct an $(1, m'_A, 0)$ block encoding of $(z + i - \hat{A} + E_0)^{-1}$ using the fast inversion of unitarily diagonalizable matrix technique in Proposition 2, which we denote by U'_A, for $|\operatorname{Im}(z)| \geqslant \eta$. We also construct an $(\alpha_B + 1, m_B + 1, 0)$ block encoding for $\hat{B} - i$ (assuming we have $(\alpha_B, m_B, 0)$ block encoding of $\hat{B}$), which can be constructed using [7, Lemma 29], and we denote it by U_B. When $\operatorname{Im}(z) < 0$ we only need to flip the sign of the extra shift.

Let $\widetilde{\sigma}_{\min}$ be a lower bound of the smallest singular value of $I + (z + i - \hat{A} + E_0)^{-1}(\hat{B} - i)$. By Theorem 1, the smallest singular value is lower bounded $1/C_{AB}$, where $C_{AB} = 1 + \|(A + B)^{-1}\|\,\|B\|$, and it is easy to check $C_{AB} \leqslant 1 + \frac{\alpha_B + 1}{\eta}$. Thus, a choice for $\widetilde{\sigma}_{\min}$ can be

$$\widetilde{\sigma}_{\min} = \frac{\eta}{\eta + \alpha_B + 1},$$

which works for all $\hat{A}$ and $\hat{B}$. However, larger values for $\widetilde{\sigma}_{\min}$ might be possible given more information about $\hat{A}$ and $\hat{B}$, as discussed in Remark 7. We can then construct a $(\frac{4}{3\widetilde{\sigma}_{\min}}, 2m'_A + m_B + 3, \epsilon'')$ block encoding of $(z - \hat{H} + E_0)^{-1}$, which uses U'_A, U_B, and other primitive gates for a total of $O(\frac{\alpha_B}{\widetilde{\sigma}_{\min}} \ln(\frac{1}{\epsilon''\widetilde{\sigma}_{\min}}))$ times.

Now we determine the complexity of Green's-function evaluation using this preconditioned solver. We apply the Hadamard test for nonunitary matrices described in Appendix D, and specifically Lemma 7, to the matrix $(z - \hat{H} + E_0)^{-1}$, for which we have constructed a block encoding using the preconditioning technique. Because amplitude estimation [47] is used in Lemma 7, we have an $1/\epsilon$ dependence on the precision rather than the $1/\epsilon^2$ often seen in Monte Carlo type methods. We also repeat amplitude estimation multiple times and take the median to exponentially reduce the failure probability of amplitude estimation, which is discussed in more detail in Appendix D.

Compared to Lemma 7, there is a further source of error due to the inexact ground energy. Suppose instead of the exact ground energy E_0 we have an approximate $\widetilde{E}_0$. Then

$$\begin{aligned}&\|(z - \hat{H} + E_0)^{-1} - (z - \hat{H} + \widetilde{E}_0)^{-1}\| \\ &\quad = |\widetilde{E}_0 - E_0|\|(z - \hat{H} + E_0)^{-1}(z - \hat{H} + \widetilde{E}_0)^{-1}\|.\end{aligned}$$

Since

$$\|(z - \hat{H} + E_0)^{-1}\| \leqslant \frac{1}{\eta}, \quad \|(z - \hat{H} + \widetilde{E}_0)^{-1}\| \leqslant \frac{1}{\eta},$$

when $|\widetilde{E}_0 - E_0| \leqslant \varsigma'$ as assumed at the beginning of this section, the error that comes from the inexact ground energy is upper bounded by ς'/η^2.

After taking into account both the error due to the block encoding of $(z - \hat{H} + E_0)^{-1}$ and the ground energy, we set $\epsilon'' = \eta\epsilon/2$ in the above analysis, and arrive at the following theorem:

Theorem 2. Given a unitary circuit U_Ψ to prepare the N-particle ground state $|\Psi_0\rangle$ to trace-norm error ς with probability at least p, a $(1/\eta, m'_A, 0)$ block encoding U'_A of $(z + i - \hat{A} + E_0)^{-1}$, an $(\alpha_B + 1, m_B + 1, 0)$ block encoding U_B of $\hat{B} - i$ for $\operatorname{Im}(z) \geqslant \eta > 0$, a lower bound $\widetilde{\sigma}_{\min}$ for the smallest singular value of $I + (z + i - \hat{A} + E_0)^{-1}(\hat{B} - i)$, and an estimate of the ground energy that has an error upper bounded by ς', we can evaluate $G_{ij}(z) = \langle\Psi_0|\hat{a}_i(z - \hat{H} + E_0)^{-1}\hat{a}_j^\dagger|\Psi_0\rangle$ to precision $\frac{8}{3\widetilde{\sigma}_{\min}}\varsigma + \frac{\varsigma'}{\eta^2} + \epsilon$ with probability δ using

(1) $O(\frac{\alpha_B}{\widetilde{\sigma}_{\min}^2\epsilon} \ln(\frac{1}{\epsilon\widetilde{\sigma}_{\min}}) \ln(\frac{1}{\delta}))$ applications of U'_A and U_B,

(2) $O(\frac{1}{\widetilde{\sigma}_{\min}\sqrt{p\epsilon}} \ln(\frac{1}{\varsigma}) \ln(\frac{1}{\delta}))$ applications of U_Ψ,

(3) other primitive gates whose number is proportional to the sum of the two numbers above.

In the absence of a tighter bound, $\widetilde{\sigma}_{\min}$ can be chosen to be $\eta/(1 + \alpha_B + \eta)$. When $\operatorname{Im}(z) \leqslant -\eta < 0$ the complexity is the same and we only need to flip the sign of the shift i in the block encodings.

C. Examples

Below we discuss the application of our preconditioned Green's-function evaluation method to the Hubbard model, the second-quantized quantum chemistry Hamiltonian in the plane-wave dual basis, and the Schwinger model, and discuss whether there is speedup compared to the direct evaluation of Green's function through QSVT or LCU as discussed in Appendix F. In all three models the Hamiltonian can be written as the sum of two terms $\hat{H} = \hat{A} + \hat{B}$, with $\|\hat{A}\| \gg \|\hat{B}\|$. The direct method using QSVT or LCU results in a linear dependence on $\|\hat{H}\|$, and therefore the dominating term, as discussed in Appendix F. Our preconditioning method

replaces the dependence on $\|\hat{H}\|$ with the dependence on the much smaller quantity $\|\hat{B}\|$. Such a dependence can, however, be cubic when no additional information about $\widetilde{\sigma}_{\min}$ in Theorem 2 is available.

1. Hubbard model

The Hubbard model can be viewed as a prototypical system for describing electrons in strongly correlated materials. The Hamiltonian shares similarity with the full Coulomb Hamiltonian in Sec. IV C 2. Consider the two-dimensional (2D) Hubbard model with $\sqrt{N} \times \sqrt{N}$ grid points that only has onsite interaction between electrons of opposite spins. The Hamiltonian reads as

$$\hat{H} = \sum_{\mathbf{x},\mathbf{y},\sigma} T(\mathbf{x}-\mathbf{y})\hat{a}^\dagger_{\mathbf{x},\sigma}\hat{a}_{\mathbf{y},\sigma} + U\sum_{\mathbf{x}} \hat{n}_{\mathbf{x},\uparrow}\hat{n}_{\mathbf{x},\downarrow} =: \hat{H}_0 + \hat{H}_1.$$

Here $\hat{n}_{\mathbf{x},\sigma} = \hat{a}^\dagger_{\mathbf{x},\sigma}\hat{a}_{\mathbf{x},\sigma}$, $\mathbf{x} = (x_1, x_2) \in \mathbb{Z}^2$, and the noninteracting part only involves nearest-neighbor interaction as

$$T(\mathbf{x}-\mathbf{y}) = \begin{cases} -t, & d(\mathbf{x},\mathbf{y}) = 1 \\ 0, & \text{otherwise.} \end{cases}$$

For most fermionic problems of interest we have $t, U > 0$. The two-dimensional domain is assumed to be periodic.

Using the fermionic Fourier transform (FFFT) [55], one may transform the creation and annihilation operators to the momentum space as

$$\hat{c}^\dagger_{\mathbf{G},\sigma} = \mathrm{FFFT}^\dagger \hat{a}^\dagger_{\mathbf{x},\sigma}\mathrm{FFFT}, \quad \hat{c}_{\mathbf{G},\sigma} = \mathrm{FFFT}^\dagger \hat{a}_{\mathbf{x},\sigma}\mathrm{FFFT},$$

where $\mathbf{G} = (G_1, G_2)$ is the index of vectors in the reciprocal space given by $G_\alpha = 2\pi k_\alpha/\sqrt{N}$, and $k_\alpha \in \{-\sqrt{N}/2 + 1, \ldots, \sqrt{N}/2\}$, $\alpha = 1, 2$. For simplicity we assume that $\sqrt{N}$ is an even number.

Explicit circuits for the FFFT, for the Jordan-Wigner fermionic representation, can be found in [55]. Using FFFT, the translation-invariant kinetic energy operator can be diagonalized in the momentum space as

$$\sum_{\mathbf{x},\mathbf{y},\sigma} T(\mathbf{x}-\mathbf{y})\hat{a}^\dagger_{\mathbf{x},\sigma}\hat{a}_{\mathbf{y},\sigma} = \mathrm{FFFT}\left(\sum_{\mathbf{G},\sigma} \hat{T}(\mathbf{G})\hat{c}^\dagger_{\mathbf{G},\sigma}\hat{c}_{\mathbf{G},\sigma}\right)\mathrm{FFFT}^\dagger. \tag{22}$$

Here $\hat{T}(\mathbf{G})$ is the Fourier transform of the discretized kinetic operator.

There are a number of ways to express such a diagonal matrix. In particular, this transformation yields $\hat{c}^\dagger_{\mathbf{G},\sigma}\hat{c}_{\mathbf{G},\sigma} = (I - Z_{\mathbf{G},\sigma})/2$, which can be simply implemented as $-Z_{\mathbf{G},\sigma}/2$ by neglecting a dynamically irrelevant shift to the energy in the equality. Applying the (ordinary) two-dimensional Fourier transform on $T(\mathbf{x})$ to find $\hat{T}(\mathbf{G})$, we find that there exists a unitary decomposition of the kinetic term such that the sum of the absolute value of the coefficients in the unitary decomposition is at most α_T which obeys

$$\alpha_T \leqslant \frac{N}{2}\max_{\mathbf{G}} |\hat{T}(\mathbf{G})| = \max_{\mathbf{G}} \left|\frac{1}{2}\sum_{\mathbf{x}} T(\mathbf{x}) e^{i\mathbf{G}\cdot\mathbf{x}}\right| \leqslant N|t|. \tag{23}$$

We further find that (again up to an irrelevant constant shift in the energy)

$$U\sum_{\mathbf{x}} \hat{n}_{\mathbf{x},\uparrow}\hat{n}_{\mathbf{x},\downarrow} = \sum_{\mathbf{x}} \frac{Z_{\mathbf{x},\uparrow}Z_{\mathbf{x},\downarrow} - Z_{\mathbf{x},\uparrow} - Z_{\mathbf{x},\downarrow}}{4}.$$

Therefore, the sum of the coefficients of the unitaries in this unitary decomposition α_U satisfies

$$\alpha_U \leqslant \sum_{\mathbf{x}} \frac{3|U|}{4} = \frac{3N|U|}{4} \leqslant N|U|. \tag{24}$$

It then follows that we can construct an $(\alpha_H, O(\ln(N/\epsilon)), 0)$ block encoding of $\hat{H}$ for

$$\alpha_H \leqslant \alpha_T + \alpha_U \leqslant N(|t| + |U|). \tag{25}$$

If $|U|$ is small compared to $|t|$, then the kinetic energy term dominates and so it makes sense to use the preconditioned algorithm to compute the Green's function where $\hat{A}$ is taken to be the kinetic operator and $\hat{B}$ the electron-electron interaction. Since the kinetic term can be diagonalized by the FFFT, we can use the fast-inversion result of Proposition 2 we can invert the kinetic part of the Hamiltonian using a single query to an oracle that yields the diagonal elements of $\hat{T}(\mathbf{G})$ and a single application of FFFT and its inverse. On the other hand, if $|t|$ is small compared to $|U|$ (which corresponds to the strongly correlated regime, and is often the case of interest) then we simply take $\hat{A}$ to be the onsite interaction term. This is similar in spirit to the hybridization expansion in quantum Monte Carlo calculations [56]. In either case, the scaling for the normalization constant is $\min(|U|, |t|)$.

We also require for our Green's-function approach an oracle that yields a block encoding of $\hat{B} + i$, where $\hat{B}$ is the two-body operator describing electron-electron interaction. As mentioned previously, a block encoding of $\hat{B}$ exists with a block-encoding factor $N\min(|t|, |U|)$. Therefore, we can construct a block encoding for $\hat{B} + i$ with a value of $\alpha_B = N\min(|t|, |U|) + 1 \in O(N\min(|t|, |U|))$. Such a block encoding requires 2 queries to an oracle that computes the energy $U\sum_{\mathbf{x}} \hat{n}_{\mathbf{x}\uparrow}\hat{n}_{\mathbf{x}\downarrow}$ for a particular configuration of electrons in position space.

The Green's function can therefore be computed with error at most ϵ and failure probability δ using a number of queries to oracles that compute the kinetic and potential energy of a given configuration (in momentum space and position space, respectively) that are of

$$\widetilde{O}\left(\frac{N^3\min(|U|, |t|)^3}{\eta^2\epsilon}\ln\left(\frac{1}{\delta}\right)\right), \tag{26}$$

which is independent of the value of $|t|$. Here we used Theorem 2 and the worst-case bound $\widetilde{\sigma}_{\min} = \eta/(1 + \eta + \alpha_B)$ as discussed in the theorem. This can provide an advantage over the nonpreconditioned version of the algorithm if $\min(|U|, |t|)$ is much smaller than $\max(|U|, |t|)$.

Remark 9. Using arguments given in Appendix J, we can further optimize this scaling to depend on the number of electrons in the initial state. If we denote the number of electrons to be N_e, then the scaling of the number of energy evaluations

is improved to

$$\widetilde{O}\left(\frac{N_e^3 \min(|U|, |t|)^3}{\eta^2 \epsilon} \ln\left(\frac{1}{\delta}\right)\right). \tag{27}$$

2. Plane-wave dual basis

For treating electrons in a periodic, infinite lattice, it is appropriate to use a periodized Coulomb operator (also called the Ewald interaction) [57]. This representation of electronic structure is also significant because it takes a similar form to the Hubbard model and consists of a kinetic and interaction term where the former can be diagonalized using the FFFT and the latter is diagonal in the computational basis. We omit the detailed discussion of the quantum many-body Hamiltonian in plane-wave dual basis here and refer readers to Ref. [55]. Unlike the Hubbard model, we find (somewhat surprisingly) that there is no clear advantage for using our preconditioned algorithm for this Hamiltonian in terms of Green's-function evaluation, at least according to our worst-case analysis.

Here we will take the U_A to be a block encoding of the external potential and two-body interaction operators and U_B to be a block encoding of the kinetic energy term. It has been found that the block-encoding factor for the kinetic energy term (denoted by α_B) is $O(\frac{N^{5/3}}{\Omega^{2/3}})$, while the block-encoding factor of the potential term (denoted by α'_A) is $O(N^{7/3}/\Omega^{2/3})$ [55]. Thus, Theorem 2 tells us that the number of applications of oracles that compute the diagonal matrix elements of the kinetic and potential operators in the plane wave and plane-wave dual basis, respectively, that are needed to compute the Green's functions within error ϵ and failure probability δ for $|\mathrm{Im}(z)| \geqslant \eta$ is in

$$\widetilde{O}\left(\frac{N^5}{\Omega^2 \eta^2 \epsilon} \ln\left(\frac{1}{\delta}\right)\right). \tag{28}$$

Again we used Theorem 2 and the worst-case bound $\widetilde{\sigma}_{\min} = \eta/(1 + \eta + \alpha_B)$ as discussed in the theorem.

Remark 10. This scaling shows that advantages do not necessarily follow by applying our technique to problems in chemistry. If we were to compare the results in Table I, then we see that the nonpreconditioned Green's-function computation scales would require a number of queries that are in $\widetilde{O}(N^{7/3}/\Omega^{2/3})$ (for constant ϵ, η, δ) (similar to the Hubbard model, we show that a small advantage can be gained by imposing a fermion-number constraint but this does not change the conclusion here). Therefore, despite the asymptotic separation between the terms, we are not able to show an advantage of preconditioned linear system solvers in the context of the plane-wave dual basis set. Therefore, preconditioned linear system solvers with better scaling with respect to α_B would be of interest for future works. It is also possible that the preconditioned solver could be more useful in different basis sets, such as the molecular orbital basis set.

3. Schwinger model

As a final example of a class of models that our methods can be applied to, we will examine computing Green's functions for the Schwinger model. The Schwinger model describes quantum electrodynamics in $1 + 1$ dimensions and is also used as a toy model for quantum chromodynamics.

The Hilbert space for the Schwinger model is the tensor product of two spaces. One consists of a tensor product of $N + 1$ fermionic spaces and the other consists of a product of N gauge-field spaces. Each gauge field can be thought of as a qubit register that can take $2L + 1$ different integer values ranging from $-L$ to L. There are two operators that we need to define that act on the gauge-field space. First we have $\hat{E}_r^2$, which is a diagonal operator and counts the energy stored in the gauge field with index $r \in \{1, \ldots, N\}$. The second is $\hat{U}_r$, which adds one to value stored in the gauge-field register and is analogous to a bosonic creation operator. The action of these operators is given formally below:

$$\hat{E}_r^2 = \sum_{\varepsilon=-L}^{L} \varepsilon^2 |\varepsilon\rangle_r\langle\varepsilon|_r, \quad \hat{U}_r = \sum_{\varepsilon=-L}^{L} |\varepsilon+1\rangle_r\langle\varepsilon|_r,$$
$$\hat{U}_r^\dagger = \sum_{\varepsilon=-L}^{L} |\varepsilon-1\rangle_r\langle\varepsilon|_r. \tag{29}$$

Here we assume for $\hat{U}_r$ and its adjoint that the gauge field satisfies periodic boundary conditions at the cutoff located at $\varepsilon = \pm L$.

The Hamiltonian can be expressed for the Schwinger model on N sites in one dimension using the operators $\hat{E}_r$ and $\hat{U}_r$ through the work of Kogut and Susskind [58] as

$$H = \sum_{r=1}^{N} \hat{E}_r^2 \otimes \hat{I}^{\otimes(N+1)} + x \sum_{r=1}^{N} [\hat{U}_r \hat{a}_r^\dagger \hat{a}_{r+1} - \hat{U}_r^\dagger \hat{a}_r \hat{a}_{r+1}^\dagger]$$
$$+ \mu \sum_{r=1}^{N} (-1)^r \hat{I}_L^{\otimes N} \hat{a}_r^\dagger \hat{a}_r. \tag{30}$$

Here we use $\hat{I}_L$ to denote the identity operation on the link variables and $\hat{I}$ to be the identity operation on the fermionic modes, where $\hat{E}_r^2$ gives the energy in the gauge field that links two sites in the one-dimensional lattice and $\hat{U}_r$ is an operator that raises or lowers excitation number for the gauge field, $\mu = 2m/(ag^2)$ and $x = 1/(ag)^2$, with a the lattice spacing, m the fermion mass, and g the coupling constant.

Once we have made this identification, we can use the construction in [59] to express the Hamiltonian as a sum of unitary operations. The simplest way to construct this as a sum of unitary matrices is to block encode $\hat{E}_r^2$ by noting $\hat{E}_r^2 = \sum_k k^2 |k\rangle\langle k|$ is block encoded by the unitary

$$\sum_k |k\rangle\langle k| \otimes e^{-iY \cos^{-1}(k^2/L^2)} :$$
$$|k\rangle|0\rangle \mapsto |k\rangle\left(\frac{k^2}{L^2}|0\rangle + \sqrt{1 - \frac{k^2}{L^2}}|1\rangle\right)$$

as per the construction that we describe in Appendix C. Similarly, $\hat{U}_r$ can be written as a sum of unitary ladder circuits. Let U'_A block encode $\sum_r \hat{E}_r^2 \otimes 1^r$ and let U_b block encode

$$x \sum_{r=1}^{N} [\hat{U}_r \hat{a}_r^\dagger \hat{a}_{r+1} - \hat{U}_r^\dagger \hat{a}_r \hat{a}_{r+1}^\dagger] + \mu \sum_r (-1)^r I_L^{\otimes r-1} \hat{a}_r^\dagger \hat{a}_r.$$

From the discussion in [59, Secs. 2.1 and 2.3] we have $\alpha_{A'} = (N-1)L^2$ and $\alpha_B = O((x+\mu)N)$.

If we allow the gauge-field cutoff to grow unboundedly, then asymptotically $\alpha_{A'}$ will dominate the complexity. If we follow the reasoning used in the previous sections, then Theorem 2, together with the worst-case bound $\widetilde{\sigma}_{\min} = \eta/(1+\eta+\alpha_B)$ as discussed in the theorem, implies that the number of queries to $U_{A'}$ and U_B are

$$\widetilde{O}\left(\frac{\alpha_B^3}{\eta^2\epsilon}\right) = \widetilde{O}\left(\frac{(x+\mu)^3N^3}{\eta^2\epsilon}\right). \tag{31}$$

There, as L increases while all other quantities remain fixed, this offers a potentially exponential improvement relative to the nonpreconditioned example. Therefore, in the simulations of quantum field theory, preconditioning solvers can significantly reduce the computational complexity with respect to the size of the cutoff.

V. FAST ALGORITHM FOR EVALUATING MATRIX FUNCTIONS

In this section we focus on implementing the matrix function $e^{-\beta H}$ where $H = A + B$, and applying it to a given quantum state $|b\rangle$. For a positive-semidefinite Hamiltonian $H = A + B \in \mathbb{C}^{N\times N}$ with $\|A\| \gg \|B\|$ and $N = 2^n$, we want to apply the imaginary-time evolution $e^{-\beta H}$. Following Sec. II B, we further assume we have access to the eigendecomposition of $A = VDV^\dagger$, where V can be efficiently implemented on a quantum computer, and D is a diagonal matrix whose entries can be queried by the following oracle:

$$O_D|k\rangle|0^r\rangle = |k\rangle|D_{kk}\rangle. \tag{32}$$

Here we assume the diagonal entries can be represented exactly by r bits. We also assume there is an $(\alpha_B, m_B, 0)$ block encoding of B denoted by U_B.

This input model is inspired by the quantum many-body Hamiltonian for which we can diagonalize the noninteracting part efficiently on a classical computer (see Sec. IV for examples). Other input models exist for different settings. For example, Ref. [39] assumes access to a block encoding of the Hamiltonian, and Ref. [44] assumes the Hamiltonian is given through a linear combination of unitaries or projections, and we have access to what is essentially the square root of the Hamiltonian $\sqrt{H}$. This allows their algorithm to achieve $O(\sqrt{\beta})$ dependence.

Table II compares the query complexities of a few different algorithms assuming we are given the block encoding of the Hamiltonian as an oracle. Note that in Table II we did not consider the dependence on β (or taking $\beta = 1$), but this dependence is included in our analysis in this section. The reason for omitting β in the table is that β is tied to the success probability of the procedure and the subnormalization of the output quantum state. When the state is normalized, the subnormalization factor amplifies the error in the process. This fact, combined with the different assumptions made in different methods, for example, Ref. [39] assumes $\beta H \gg I$, complicates the fair comparison of different methods.

The rest of the section is organized as follows. We introduce an algorithm based on the contour-integral formulation in Sec. V A, and a different algorithm based on the inverse transform in Sec. V B. Both formulations can be used to prepare a purified Gibbs state, which is discussed in Sec. V C.

A. Contour-integral formulation

We use the contour-integral representation

$$e^{-\beta x} = \frac{1}{2\pi i}\oint_\Gamma \frac{e^{-\beta z}}{z-x}dz, \tag{33}$$

where $x \geqslant 0$ and the contour is chosen as

$$\Gamma = \{t^2 - \zeta + it \in \mathbb{C} : t \in \mathbb{R}\}. \tag{34}$$

We choose the parameter ζ in the following way, and will explain the reason in Appendix G:

$$b = \min\left(\frac{1}{2\beta}, \frac{1}{6}\right), \quad \zeta = 2b(1-b). \tag{35}$$

In particular, $\beta\zeta \leqslant 1$. Equation (33) then enables us to express the matrix function $e^{-\beta H}$ in terms of a linear combination of matrix inverses:

$$e^{-\beta H} = \frac{1}{2\pi i}\oint_\Gamma e^{-\beta z}(z-H)^{-1}dz. \tag{36}$$

The contour-integral formulation has been widely used to compute matrix functions on classical computers [18]. In Ref. [60], a number of parabolic contours have been considered, which are optimally tuned so that fast exponential convergence can be reached with respect to the number of discretization points N. These types of discretized contour integrals have also been used to obtain rational approximation [61], and to invert Laplace transform [62–64]. However, the parabolas they use for $e^{-\beta x}$ move away from the origin in the negative direction along the real axis, and from our analysis in this section we find that this results in an exponentially growing subnormalization factor in the LCU procedure. Therefore, we need to design new parametrization of contours in Eq. (35).

Once the contour is chosen, we may truncate the contour Γ on a finite interval $t \in [-T, T]$, and apply the Gauss-Legendre quadrature formula to discretize this truncated contour integral. This leads to

$$e^{-\beta x} \approx \sum_{j\in[J]} \frac{\varrho_j}{z_j - x}, \tag{37}$$

where

$$z_j = t_j^2 - \zeta + it_j, \quad \varrho_j = \frac{T}{2\pi i}w_j e^{-\beta z_j}(2t_j + i), \quad t_j = Ts_j, \tag{38}$$

and s_j, w_j are the nodes and weights of Gauss-Legendre quadrature, respectively, and the truncation range $T \geqslant 1$ is to be chosen. The contour and the quadrature points are shown in Fig. 3. According to Appendix G, the truncation error decays very rapidly as T increases and therefore we do not need to choose a large T. We first bound the error of this approximation in the following lemma.

Lemma 2. With z_j and ϱ_j defined above, and $\widetilde{\beta} = \min(\beta, 3)$, we have

$$\left|e^{-\beta x} - \sum_{j\in[J]}\frac{\varrho_j}{z_j - x}\right| \leqslant \sqrt{\frac{2}{\beta\pi}}e^{1-\beta T^2} + \frac{64T^2\widetilde{\beta}e^{3/2}}{1-e^{-1/(8T\widetilde{\beta})}}e^{-J/(4T\widetilde{\beta})},$$

for all $x \geqslant 0$.

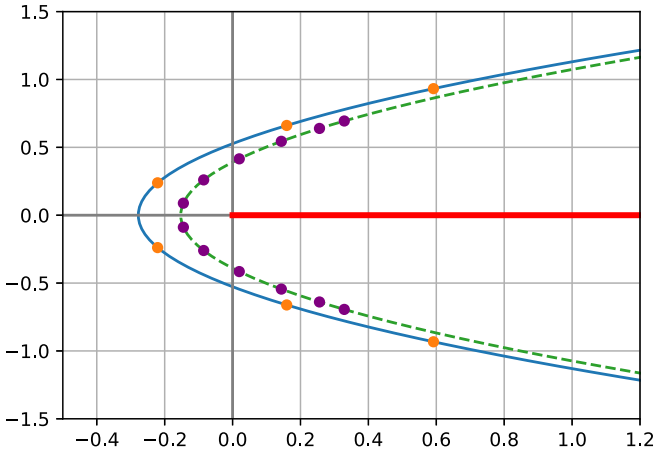


FIG. 3. The parabolic contours Γ and Gauss-Legendre quadrature nodes $\{z_j\}$. The parabola with solid line is for $\beta = 3$ and the one with dashed line is for $\beta = 6$. The spectrum of the Hamiltonian $H = A + B$ is on the positive part of the real axis (solid red line). The dots on the parabolas show the quadrature nodes. Unlike the usual contour integral, this parabolic contour extends to the infinity. This is, however, not a problem because the integrand decays very quickly to 0.

The proof of this lemma is in Appendix G. As can be seen here, as we increase the truncation range T and number of quadrature nodes J the error decays to zero.

With the approximation in Eq. (37), we can implement $e^{-\beta H}$ as a linear combination of $(z_j - H)^{-1}$ by

$$e^{-\beta H} \approx \sum_{j\in[J]} \varrho_j (z_j - H)^{-1}. \tag{39}$$

To do this, we first need the block encoding of the following matrix (called a select oracle in the context of LCU):

$$\begin{aligned} S &= \sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j - H)^{-1} \\ &= \left(\sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j + \xi_j - A) - \sum_{j\in[J]} |j\rangle\langle j| \otimes (B + \xi_j) \right)^{-1}, \end{aligned} \tag{40}$$

where we choose

$$\xi_j = \begin{cases} i, & \text{if } \operatorname{Im} z_j > 0 \\ -i, & \text{if } \operatorname{Im} z_j \leqslant 0. \end{cases} \tag{41}$$

We can see that the operator inside the inverse, which can be seen as a block-diagonal matrix, is the sum of two parts, with the operator norm of the first part being much larger than that of the second part. Therefore, we may employ the preconditioned linear system solver.

Since we have access to the eigendecomposition of A, we can obtain the various block encodings needed in Theorem 1 easily. We summarize the cost for constructing these block encodings in the following lemma.

Lemma 3. (a) A $(1, m_1, 0)$ block encoding of $(\sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j + \xi_j - A))^{-1}$ can be implemented by $O(1)$ applications of V, O_D, and their inverses, with $m_1 = O(\text{poly}(r + \ln(J)))$. (b) A $(1 + \alpha_B, m_2, 0)$ block encoding of $\sum_{j\in[J]} |j\rangle\langle j| \otimes (B + \xi_j)$ can be implemented by one application of U_B, with $m_2 = O(\text{polylog}(J))$.

We will provide the construction in Appendix H. Here the polylog factors originate from the classical computation related to $\{z_j, \xi_j\}$, as discussed in Remark 3. We further have the following bounds:

$$\begin{aligned} &\left\| \left(\sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j + \xi_j - A) \right)^{-1} \right\| \leqslant 1, \\ &\|(z_j - A - B)^{-1}\| \leqslant \frac{1}{\zeta} \leqslant 2 \max(\beta, 3), \\ &\left\| \sum_{j\in[J]} |j\rangle\langle j| \otimes (B_j + \xi_j) \right\| \leqslant 1 + \alpha_B. \end{aligned} \tag{42}$$

We introduce the parameter $\widetilde{\sigma}'_{\min}$ to be a lower bound of the smallest singular values of $I + (z_j + \xi_j - A)^{-1}(B_j + \xi_j)$ for all j. In other words,

$$1/\widetilde{\sigma}'_{\min} \geqslant \max_{j\in[J]} \|[I + (z_j + \xi_j - A)^{-1}(B_j + \xi_j)]^{-1}\|. \tag{43}$$

We want to invert the block-diagonal matrix $\sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j - A - B)^{-1}$, and therefore $\widetilde{\sigma}'_{\min}$ plays the same role as $\widetilde{\sigma}_{\min}$ in Theorem 1. We would prefer $\widetilde{\sigma}'_{\min}$ to be a tight lower bound in order to save computational cost. By Lemma 1, when no better bound is available, we can choose

$$\widetilde{\sigma}'_{\min} = 1/\max_{j\in[J]} [1 + \|(z_j - A - B)^{-1}\| \, \|B\|] = \Omega(1/\beta\alpha_B).$$

Using these block encodings and bounds, by Theorem 1, we have the following lemma:

Lemma 4. Let $\widetilde{\sigma}'_{\min}$ satisfy Eq. (43). An $(\alpha_S, m_S, \epsilon')$ block encoding of the operator S defined in Eq. (4) can be implemented using $O(\frac{\alpha_B}{\widetilde{\sigma}'_{\min}} \ln(\frac{1}{\widetilde{\sigma}'_{\min}\epsilon'}))$ applications of V, O_D, U_B, and their inverses, where $\alpha_S = O(1/\widetilde{\sigma}'_{\min})$ and $m_S = O(\text{poly}(r + \ln(J)))$. Furthermore, it is guaranteed that $\widetilde{\sigma}'_{\min} = \Omega(1/(\beta\alpha_B))$.

This block encoding of the operator S, with some unitaries to prepare a state containing the coefficients

$$|c\rangle = \frac{\sum_j \sqrt{|\varrho_j|}|j\rangle}{\sum_j |\varrho_j|}, \quad |c'\rangle = \frac{\sum_j \sqrt{|\varrho_j|} e^{i\theta_j} |j\rangle}{\sum_j |\varrho_j|}, \tag{44}$$

where the phase factor θ_j satisfies $\varrho_j = |\varrho_j| e^{i\theta_j}$. This enables us to perform the LCU procedure for $\sum_j \varrho_j (z_j - H)^{-1}$ through

$$\begin{aligned} &(\langle c| \otimes I_n) \left(\sum_{j\in[J]} |j\rangle\langle j| \otimes (z_j - H)^{-1} \right) (|c'\rangle \otimes I) \\ &= \sum_{j\in[J]} \varrho_j (z_j - H)^{-1}. \end{aligned}$$

What we get in the end is an $(\alpha_{\text{LCU}}, m_{\text{LCU}}, \epsilon')$ block encoding of $\sum_j \varrho_j (z_j - H)^{-1}$, where

$$\alpha_{\text{LCU}} = \alpha_S \sum_j |\varrho_j|. \tag{45}$$

Note that [using Eq. (38)]

$$\sum_j |\varrho_j| = \frac{T}{2\pi}\sum_j w_j |e^{-\beta z_j}(2t_j+1)| \to \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{-\beta(t^2-\zeta)}(2|t|+1) = O\left(\frac{1}{\sqrt{\beta}}\right), \tag{46}$$

where we have used the relation $\beta\zeta \leqslant 1$. Therefore, substituting this into Eq. (45) we have $\alpha_{\rm LCU} = O(\alpha_B\sqrt{\beta})$.

We then estimate the error of this block encoding. So far, the error for block encoding $\sum_j \varrho_j (z_j - H)^{-1}$ has been accounted for, and it is bounded by ϵ'. An additional source of error is due to the difference between $\sum_j \varrho_j (z_j - H)^{-1}$ and $e^{-\beta H}$. This error is bounded by Lemma 2. The total error ϵ is the sum of these two errors. Therefore, we set $\epsilon' = \epsilon/2$, and choose J and T so that the error bound in Lemma 2 is bounded by $\epsilon/2$. We need to choose T and J to be

$$T = O\left(\sqrt{\max\left(1, \frac{1}{\beta}\right)\ln\left(\frac{1}{\epsilon}\right)}\right),$$
$$J = \widetilde{O}\left(\max(1,\beta)\left[\ln\left(\frac{1}{\epsilon}\right)\right]^{3/2}\right). \tag{47}$$

The above results can be summarized into the following theorem.

Theorem 3. Let $\widetilde{\sigma}'_{\rm min}$ satisfy Eq. (43). An $(\alpha_{\rm LCU}, m_{\rm LCU}, \epsilon)$ block encoding can be constructed for $e^{-\beta H}$, with $H = A + B$ as described at the beginning of this section and the oracles V, O_D, and U_B described above, where

$$\alpha_{\rm LCU} = O(1/(\sqrt{\beta}\widetilde{\sigma}'_{\rm min})),$$
$$m_{\rm LCU} = O(\text{poly}(r + \ln(\beta) + \ln\ln(1/\epsilon))),$$

using $O(\frac{\alpha_B}{\widetilde{\sigma}'_{\rm min}}\ln(\frac{1}{\widetilde{\sigma}'_{\rm min}\epsilon'}))$ applications of V, O_D, U_B, and their inverses. Furthermore, it is guaranteed that $\widetilde{\sigma}'_{\rm min} = \Omega(1/(\beta\alpha_B))$.

We remark that the number of qubits needed is $n + O(\text{poly}(r + \ln(\beta) + \ln\ln(1/\epsilon)))$ where the poly comes from the cost of classical Boolean circuits to perform algebraic operations on eigenvalues of A and the conversion to quantum circuit using [46, Lemma 10.10]. A more detailed estimate of the number of qubits needed and the gate complexity will require estimating these costs.

Given the block encoding we can apply the matrix function $e^{-\beta H}$ to a quantum state $|b\rangle$ which is prepared by a circuit U_b, i.e., $U_b|0^n\rangle = |b\rangle$. We let $\xi = \|e^{-\beta H}|b\rangle\|$, and the goal is to prepare a state $e^{-\beta H}|b\rangle/\xi$. Directly applying the block encoding has a success probability of $\Omega(\xi^2/\alpha_{\rm LCU}^2)$. Using amplitude amplification [47] we can boost the success probability to at least $\frac{1}{2}$ with $O(\alpha_{\rm LCU}/\xi)$ applications of the block-encoding circuit and its inverse. Because of the subnormalization of the output quantum state by a factor of ξ, we need the block-encoding error to be $O(\xi\epsilon)$ instead of $O(\epsilon)$. Therefore, in total we need to query V, O_D, U_B, and their inverses $O(\frac{\alpha_B}{\xi\sqrt{\beta}\widetilde{\sigma}'^2_{\rm min}}\ln(\frac{1}{\xi\epsilon\widetilde{\sigma}'_{\rm min}}))$ times, and U_b and its inverse $O(\frac{1}{\xi\sqrt{\beta}\widetilde{\sigma}_{\rm min}})$ times. Therefore, we have obtained the complexity for applying matrix exponential using contour integral in Table II.

B. Inverse transform formulation

The basic idea of using the inverse transform to accelerate the computation of $f(A+B)$ is as follows. We assume that $(A+B) \succ 0$ and that $(A+B)^{-1}$ can be efficiently block encoded with a block-encoding factor α'_{A+B}, as indicated in Theorem 1. The inverse transform is simply

$$f(A+B) = g((A+B)^{-1}), \tag{48}$$

where

$$g(y) = f(y^{-1}). \tag{49}$$

Then, instead of finding a block encoding of $f(A+B)$, we can alternatively find a block encoding for $g((A+B)^{-1})$. The efficiency of such a transformation relies on the behavior of g within the interval $[-1, 1]$. In particular, the behavior of g near the origin plays a critical role, which reflects the decay of the original function f at infinity. Our strategy is then to approximate $g(y)$ uniformly on $[-1, 1]$ by a Chebyshev series, and the truncated Chebyshev series can then be implemented via QSVT. Compared to the contour-integral formulation, the use of the inverse transform does not require the usage of the LCU technique, and provides a simpler method for preparing the thermal state.

An example is the imaginary-time evolution $e^{-\beta H}$. The corresponding function is $f(x) = \text{sgn}(x)e^{-\zeta x}$, and we may construct $g(y) = \text{sgn}(y)e^{-\zeta|y|^{-1}}$. The parameter ζ will later be specified to be $\zeta = \beta/\alpha'_{A+B}$ to block encode $e^{-\beta H} = g((A+B)^{-1}/\alpha'_{A+B})$. The reason why we put a sgn function in $f(x)$ and $g(y)$ is to ensure that $g(y)$ is an odd function on $[-1, 1]$, then the corresponding truncated Chebyshev series is also odd and can be implemented via QSVT discussed in Appendix B, in which we only describe how to apply QSVT to block encode odd polynomials for technical simplicity. We remark that QSVT can also be used to block encode general polynomials (see Remark 11), therefore, the inverse transform approach can be applied to general functions, provided that the function $g(y)$ is in the Gevrey class (to be defined later), which means that the original function $f(x)$ decays at infinity in some sense.

We first note that the function $\text{sgn}(y)e^{-\zeta|y|^{-1}}$ is in $C^\infty([-1, 1])$ but not real analytic at $y = 0$. This complicates the convergence analysis when we approximate this function via polynomials of y. Nonetheless, we shall show that the query complexity only scales logarithmically with respect to the error ϵ. To this end, we first define a subset of smooth functions, called the Gevrey class as follows.

Definition 3. The Gevrey class of order σ on an interval $\mathcal{I}$ is defined as

$$G^\sigma(\mathcal{I}) = \{g(y) \in C^\infty(\mathcal{I}) : \exists C > 0, R > 0, s.t. |g^{(k)}(x)| \leqslant CR^k(k!)^\sigma,\ \forall\, x \in \mathcal{I}, k \geqslant 0\}. \tag{50}$$

Note that $G^1([-1, 1])$ represents the class of real analytic functions on $[-1, 1]$, and $G^1([-1, 1]) \subset G^\sigma([-1, 1]) \subset C^\infty([-1, 1])$ for all $\sigma > 1$. In order to show that $\text{sgn}(y)e^{-\zeta|y|^{-1}}$ belongs to the Gevrey class, we will use the chain rule of high-order derivatives (called the Faà di Bruno's formula, see e.g. [65, Theorem 1.3.2]):

Lemma 5 (Faà di Bruno's formula). Let h, g be smooth functions, and $f(s) = h(g(s))$, then

$$f^{(k)}(s) = \sum \frac{k!}{q_1!(1!)^{q_1} q_2!(2!)^{q_2} \dots q_k!(k!)^{q_k}} \times h^{(q_1+q_2+\cdots+q_k)}(g(s)) \prod_{j=1}^{k} (g^{(j)}(s))^{q_j}, \tag{51}$$

where the sum is taken over all k-tuples of non-negative integers $(q_1, \dots, q_k)$ satisfying $\sum_{j=1}^{k} j q_j = k$.

Proposition 4. Let $g(y) = \mathrm{sgn}(y) e^{-\zeta |y|^{-1}}$. Then for any $\zeta > 0$, $g(y) \in G^3([-1, 1])$, with $C = 1, R = 16e\zeta^{-1}$.

Proof. Without loss of generality we only consider $y > 0$. We view $e^{-\zeta y^{-1}}$ as the composition of e^w and $w = -\zeta y^{-1}$. Then, by Lemma 5,

$$\begin{aligned} g^{(k)}(s) &= \sum_{\sum j q_j = k} \frac{k!}{q_1!(1!)^{q_1} q_2!(2!)^{q_2} \dots q_k!(k!)^{q_k}} \\ &\quad \times e^{-\zeta y^{-1}} \prod_{j=1}^{k} [(-1)^{j+1} \zeta j! y^{-j-1}]^{q_j} \\ &= \sum_{\sum j q_j = k} \frac{k!}{q_1! q_2! \dots q_k!} e^{-\zeta y^{-1}} (-1)^{k+\sum q_j} \zeta^{\sum q_j} y^{-k-\sum q_j}. \end{aligned}$$

Using the substitution $z = \zeta y^{-1}$ and the fact that the function $e^{-z} z^m$ achieves its maximum at $z = m$, we have

$$\begin{aligned} e^{-\zeta y^{-1}} \zeta^{\sum q_j} y^{-k-\sum q_j} &= \zeta^{-k} e^{-z} z^{k+\sum q_j} \\ &\leqslant \zeta^{-k} e^{-k-\sum q_j} \Big(k + \sum q_j\Big)^{k+\sum q_j} \\ &\leqslant \left(\frac{4}{e\zeta}\right)^k (k^k)^2 \leqslant \left(\frac{4e}{\zeta}\right)^k (k!)^2, \end{aligned}$$

where the last step is due to the inequality $k^k \leqslant e^k k!$. Then we have

$$|g^{(k)}(s)| \leqslant \left(\frac{4e}{\zeta}\right)^k (k!)^2 \sum_{\sum j q_j = k} \frac{k!}{q_1! q_2! \dots q_k!}.$$

Finally, by directly loosing $\frac{k!}{q_1! q_2! \dots q_k!}$ to $k!$ and noticing that the number of tuples $(q_1, \dots, q_k)$ satisfying $\sum j q_j = k$ is less than $(k+1)(k/2+1)\dots(k/k+1) = \binom{2k}{k} < 2^{2k}$, we have

$$|g^{(k)}(s)| \leqslant \left(\frac{4e}{\zeta}\right)^k (k!)^2 2^{2k} k! = \left(\frac{16e}{\zeta}\right)^k (k!)^3.$$

■

Now let us consider the convergence analysis of the Chebyshev polynomial expansion of $g(y)$. We remark that the proof based on the contour-integral formulation is only possible if the function is complex analytic in a neighborhood of $[-1, 1]$ in the complex plane, as shown in the proof of Lemma 2. This is not satisfied in the case of the inverse transform. Here we present another approach following, e.g., [66, Sec. 5.7]. The proof of Theorem 4 is given in Appendix I.

Theorem 4. Let $g(y) \in C^{r+1}([-1, 1])$, and $T_k(\cdot)$ be the kth-order Chebyshev polynomial on $[-1, 1]$. Let

$$c_k = \frac{2 - \delta_{k0}}{\pi} \int_{-1}^{1} \frac{g(y) T_k(y)}{\sqrt{1 - y^2}} dy, \quad k \geqslant 0. \tag{52}$$

Then for $d \geqslant 1$

$$\|g(y) - \sum_{k=0}^{d} c_k T_k(y)\|_\infty \leqslant 32 \frac{8^r (r+1)! \|g^{(r+1)}\|_\infty}{d^r}. \tag{53}$$

If the function $g(y)$ is smooth, we can choose r to be arbitrarily large to obtain better convergence with respect to d. However, if the derivatives of g also grow rapidly when r increases, the error will eventually grow up again. Therefore, we may choose an optimal r to balance the increasing norm of higher-order derivatives and better convergence order to obtain smallest error. Under the assumption of Gevrey class, we have the following result.

Theorem 5. Assume $g(y) \in G^\sigma([-1, 1])$ for some $\sigma \geqslant 0$. Then for any $\epsilon > 0$, to achieve an ϵ approximation of $g(y)$, *i.e.,* $\|g(y) - \sum_{k=0}^{d} c_k T_k(y)\|_\infty \leqslant \epsilon$, it suffices to choose

$$d \geqslant 8eR[\ln(32CR/\epsilon) + 2]^{\sigma+1}.$$

Proof. By Theorem 4 and the definition of Gevrey class,

$$\|g(y) - \sum_{k=0}^{d} c_k T_k(y)\|_\infty \leqslant 32C \frac{8^r R^{r+1} [(r+1)!]^{\sigma+1}}{d^r} \tag{54}$$

holds for all $r \geqslant 1$. To achieve an ϵ approximation, it then suffices to choose m such that

$$32C \frac{8^r R^{r+1} [(r+1)!]^{\sigma+1}}{d^r} \leqslant \epsilon, \tag{55}$$

which is equivalently

$$d \geqslant \frac{8(32C)^{\frac{1}{r}} R^{1+\frac{1}{r}} [(r+1)!]^{\frac{\sigma+1}{r}}}{\epsilon^{\frac{1}{r}}}. \tag{56}$$

Using the estimate $(r+1)! \leqslant (r+1)^r$, it suffices to choose

$$d \geqslant 8R \left(\frac{32CR}{\epsilon}\right)^{\frac{1}{r}} (r+1)^{\sigma+1}. \tag{57}$$

Now choose $r = \lceil \ln(32CR/\epsilon) \rceil$, then the sufficient condition for d becomes

$$d \geqslant 8eR \left[\ln\left(\frac{32CR}{\epsilon}\right) + 2\right]^{\sigma+1}. \tag{58}$$

We complete the proof. ■

Finally, to implement the truncated Chebyshev series $\sum_{k=0}^{d} c_k T_k(A)$ for some Hamiltonian A, we may use QSVT (see Appendix B). Let us now consider the complexity to construct a block encoding for $\frac{1}{2} e^{-\beta(A+B)}$ for $(A+B) \succ 0$. The reason for adding a subnormalization constant $\frac{1}{2}$ is to ensure that the corresponding Chebyshev polynomial is bounded by 1, which allows to use QSVT approach for block encoding.

Theorem 6. Let $\widetilde{\sigma}_{\min}$ be a lower bound for the smallest singular value of $I + A^{-1}B$. Let U_A' be an $(\alpha_A', m_A', 0)$ block encoding of A^{-1}, U_B be an $(\alpha_B, m_B, 0)$ block encoding of B. Then for any $\beta > 0$ and $0 < \epsilon < 1$, a $(1, m_{\mathrm{QSVT}}, \epsilon)$ block

encoding can be constructed for $\frac{1}{2}e^{-\beta H}$ with $H = A + B \succ 0$, where

$$m_{\mathrm{QSVT}} = 2m'_A + m_B + 4,$$

with the following costs:

(1) $\widetilde{O}(\frac{\alpha'^2_A \alpha_B}{\beta \widetilde{\sigma}^2_{\min}}[\ln(\frac{1}{\epsilon})]^5)$ applications of U'_A, U_B, their controlled versions, and their inverses,

(2) $(n + 2m'_A + m_B + 4)$ qubits,

(3) $O((2m'_A + m_B + 4)\frac{\alpha'_A}{\beta \widetilde{\sigma}_{\min}}[\ln(\frac{1}{\epsilon})]^4)$ additional primitive quantum gates due to the QSVT approach.

Furthermore, $\widetilde{\sigma}_{\min} = \Omega(1/[1 + \|(A + B)^{-1}\| \, \|B\|])$.

Proof. By Theorem 1, for any $\epsilon' > 0$ (to be specified later), we can construct an $(\alpha'_{A+B}, m'_{A+B}, \epsilon')$ block encoding of $(A + B)^{-1}$, with $\alpha'_{A+B} = \frac{4\alpha'_A}{3\widetilde{\sigma}_{\min}}$, $m'_{A+B} = 2m'_A + m_B + 3$, using $O(\frac{\alpha'_A \alpha_B}{\widetilde{\sigma}_{\min}} \ln(\frac{\alpha'_A}{\epsilon' \widetilde{\sigma}_{\min}}))$ applications of U'_A, U_B, their controlled versions, their inverses, and other primitive gates.

Consider the function $g(y) = \frac{1}{2}\,\mathrm{sgn}(y) e^{-\beta \alpha'^{-1}_{A+B}|y|^{-1}}$. By Theorem 5 and Proposition 4 (here $C = \frac{1}{2}$ and $R = 16e\beta^{-1}\alpha'_{A+B}$), there exists a d-degree odd polynomial $P(y)$ with $d = \Theta(\alpha'_{A+B}\beta^{-1}[\ln(\alpha'_{A+B}\beta^{-1}\epsilon^{-1})]^4)$, such that

$$\left\| \frac{1}{2} e^{-\beta(A+B)} - P\left(\frac{(A+B)^{-1}}{\alpha'_{A+B}}\right) \right\| = \left\| g\left(\frac{(A+B)^{-1}}{\alpha'_{A+B}}\right) - P\left(\frac{(A+B)^{-1}}{\alpha'_{A+B}}\right) \right\| \leqslant \frac{\epsilon}{2}.$$

Note that $\|g(\frac{(A+B)^{-1}}{\alpha'_{A+B}})\| \leqslant \frac{1}{2}$. Therefore, for any $\epsilon < 1$, we always have $\|P(\frac{(A+B)^{-1}}{\alpha'_{A+B}})\| < 1$. By QSVT and the block encoding of $(A + B)^{-1}$, we can construct a $(1, m'_{A+B} + 1, 2d\epsilon')$ block encoding of $P(\frac{(A+B)^{-1}}{\alpha'_{A+B}})$, using d queries of the block encoding of $(A + B)^{-1}$ and its inverse, and $O((m'_{A+B} + 1)d)$ additional primitive gates. To control the total error by ϵ from above, it suffices to choose $\epsilon' = \epsilon/(4d)$. Plugging this back into the complexity analysis and multiplying them in all the steps, we obtain the estimates of the costs as stated in the theorem. ∎

Finally, let us consider applying the matrix function $e^{-\beta H}$ to a quantum state $|b\rangle$, i.e., preparing a state $e^{-\beta H}|b\rangle/\xi$ with $\xi = \|e^{-\beta H}|b\rangle\|$. Similar to the discussion in Sec. V A, the number of queries to V, O_D, U_B, and their inverses scales $\widetilde{O}(\frac{\alpha'^2_A \alpha_B}{\beta \xi \widetilde{\sigma}^2_{\min}}[\ln(\frac{1}{\epsilon})]^5)$, and the number of queries to U_b and its inverse becomes $O(\frac{1}{\xi}\ln(\frac{1}{\epsilon}))$.

C. Purified Gibbs state

The Gibbs state is a mixed state whose density matrix is

$$\frac{1}{Z_\beta} e^{-\beta H},$$

where $Z_\beta = \mathrm{Tr}(e^{-\beta H})$ is the partition function. In this section we assume $H = A + B$ with the A and B accessed through the oracles outlined at the beginning of Sec. V.

The Gibbs state can be constructed as a partial trace of a pure state, called the purified Gibbs state:

$$|\Psi\rangle = \frac{1}{\sqrt{Z_\beta}} \sum_{x \in [N]} |x\rangle (e^{-\beta H/2}|x\rangle). \tag{59}$$

Here x enumerates the elements of the n-qubit computational basis. When we trace out the first register in the density matrix $|\Psi\rangle\langle\Psi|$, we will obtain the Gibbs state. These states are important for any algorithm that seeks to use thermal states and gain advantages from amplitude amplification, such as recent results on semidefinite programming as well as results on training Boltzmann machines [20,39,67]. We may rewrite $|\Psi\rangle$ as

$$|\Psi\rangle = \sqrt{\frac{N}{Z_\beta}} (I \otimes e^{-\beta H/2}) \left(\frac{1}{\sqrt{N}} \sum_{x \in [N]} |x\rangle|x\rangle \right).$$

Therefore, we can prepare the purified Gibbs state by applying the matrix function $I \otimes e^{-\beta H/2}$ to the maximally entangled state $(1/\sqrt{N})\sum_x |x\rangle|x\rangle$ [38], which in turn can be prepared by applying a Hadamard transform and a series of CNOTgates similar to how one prepares the EPR pair [48].

From the discussion at the end of Secs. V A and V B, since we have $\xi = \sqrt{Z_\beta/N}$, the query complexity of preparing the purified Gibbs state is as follows:

Corollary 2. In order to prepare the purified Gibbs state, the total number of queries to U'_A, O_D, U_B and their inverses is $\widetilde{O}((\alpha_B/\sqrt{\beta}\widetilde{\sigma}'^2_{\min})\ln(1/\epsilon)\sqrt{N/Z_\beta})$ in the contour-integral approach, and $\widetilde{O}((\alpha'^2_A \alpha_B/\beta\widetilde{\sigma}^2_{\min})[\ln(1/\epsilon)]^5\sqrt{N/Z_\beta})$ in the inverse transform approach, where $\widetilde{\sigma}'_{\min}$ satisfies Eq. (43) and $\widetilde{\sigma}_{\min}$ is a lower bound for the smallest singular value of $I + A^{-1}B$.

Note that if we only care about the case when $\beta = 1$ and $A + B \succ 0$, we can shift A such that the assumptions $A \succeq I$ and $A + B \succeq I$ hold without introducing much overhead in the subnormalization constant, then $\alpha'_A = O(1)$ and the query complexity becomes $\widetilde{O}((\alpha_B/\widetilde{\sigma}^2_{\min})[\ln(1/\epsilon)]^5\sqrt{N/Z_\beta})$, as shown in Table II.

VI. DISCUSSION

We have a quantum primitive called fast inversion to solve a class of quantum linear systems problem (QLSP) $|x\rangle \propto A^{-1}|b\rangle$. If A is a normal matrix and diagonalized as $A = VDV^\dagger$, then fast inversion is applicable if (1) there is an efficient quantum circuit to implement the unitary matrix V (for instance, when V can be implemented via the quantum Fourier transform), and (2) the inverse of the diagonal matrix D^{-1} can be efficiently implemented via a classical circuit. Here, compared to the standard approach of first finding a block encoding of A and then invert A, the condition (2) is the key reason for fast inversion to achieve reduced circuit depth and query complexities.

Using fast inversion, we developed a preconditioned linear system solver for solving a class of QLSP of the form $|x\rangle \propto (A + B)^{-1}|b\rangle$. Here we assume the matrix A can be fast inverted, but $\|A\| \gg \|B\|$, $\|A^{-1}\|$, $\|(A + B)^{-1}\|$. Our main result is that the query complexity of the preconditioned linear system solver can be independent of $\|A\|$, or the condition number $\kappa(A + B)$, and can therefore significantly reduce the computational cost.

We demonstrate an application of the preconditioned quantum linear system solver for computing the single-particle Green's function in quantum many-body physics. Although quantum linear system solver is often considered to be used as a subroutine of a larger quantum algorithm, the computation of Green's functions in quantum many-body physics is entirely a linear system problem in high dimensions. Hence, we need to take into account all components of the algorithm, in particular the readout errors due to the Monte Carlo sampling. Using fast inversion, we again demonstrate that the query complexity can be independent of the norm of the total Hamiltonian.

Observe that $(A+B)^{-1}|b\rangle$ is only one example of a more general class of problems of computing $f(A+B)|b\rangle$, where $f(A+B)$ is a matrix function. To be specific, we consider the problem of thermal state preparation, where $f(H)=e^{-\beta H}$ and $H=A+B$ is a Hermitian matrix. We again assume that the main difficulty comes from that $\|H\| \sim \|A\|$ is very large. We developed two methods to approximately compute the matrix function via a series of linear systems. The first method is based on Cauchy's contour-integral formula, and the second is based on a simple inverse transform. Using fast inversion, both methods can be used to fast prepare a thermal state, and the cost is independent of $\|H\|$. We remark that both the contour-integral formula and the inverse transform are quite general, and can be used to accelerate the computation of other matrix functions as well.

We would like to remark that fast inversion is intimately connected to the fast forwarding of Hamiltonian simulation $e^{iA\tau}$, where A is Hermitian and $\tau \in \mathbb{R}$ is some arbitrary time. For example, based on fast forwarding, the interaction picture Hamiltonian simulation algorithm [68] allows fast evaluation of $e^{i(A+B)t}|b\rangle$. The assumption and the main result in [68] are both similar to this paper, i.e., when $\|A\| \gg \|B\|$ and $e^{iA\tau}$ can be fast forwarded, the query complexity of the Hamiltonian simulation $e^{i(A+B)t}|b\rangle$ can be independent of $\|A\|$. It is interesting to compare fast inversion and fast forwarding from a numerical analysis perspective, i.e., whether one can use fast inversion to perform fast forwarding, and vice versa. More generally, when considering the fast evaluation of certain matrix functions $f(A+B)$, whether fast inversion or fast forwarding can lead to a more efficient algorithm. The answer will likely depend on the details of the function f of interest, as well as the preconstants of different methods. This will be our future work.

ACKNOWLEDGMENTS

This work was partially supported by the Department of Energy under Grants No. DE-AC02-05CH11231 (L.L. and Y.T.), No. DE-SC0017867 (D.A. and L.L.), the Google Quantum Research Award, Quantum Algorithms, Software, and Architectures (QUASAR) Agile Investment at Pacific Northwest National Laboratory, the Pacific Northwest Laboratory LDRD program (N.W.), and the National Science Foundation under the QLCI program through Grant No. OMA-2016245 (L.L. and N.W.).

APPENDIX A: QUANTUM LINEAR SYSTEM SOLVERS

We first briefly review the literature for solving QLSPs of the form $A|x\rangle \propto |b\rangle$ without preconditioning. Here, we consider a block-encoding model [7], i.e., there exists a unitary matrix U_A that encodes the matrix A (see Sec. I B for the details of block encoding), while the right-hand side vector $|b\rangle$ is prepared by a unitary U_b as $|b\rangle = U_b|0^n\rangle$. For simplicity, we only compare the query complexity to the block encoding U_A.

The query complexity of the original quantum linear systems algorithm by Harrow-Hassidim-Lloyd (HHL) algorithm [8][2] scales as $\widetilde{O}(\kappa^2/\epsilon)$, where $\kappa=\kappa(A)$ and ϵ is the target accuracy. The HHL algorithm is based on the phase estimation method. In the past few years, there have been significant progresses towards reducing the query complexity for quantum linear solvers. In particular, the linear combination of unitaries (LCU) [6,36] and quantum singular value transformation (QSVT) [7,45] (closely related to quantum signal processing (QSP) [37,69]) techniques can reduce the query complexity to $O(\kappa^2 \mathrm{polylog}(\kappa/\epsilon))$. It is worth commenting that with respect to the condition number, the worst-case complexity $\widetilde{O}(\kappa^2)$ is very much inherent to all aforementioned algorithms (HHL, LCU, QSVT). This is because the cost of each algorithm is $\widetilde{O}(\kappa)$ per run, and the worst-case success probability of each run is $\widetilde{O}(\kappa^{-2})$. Hence, in order to boost to $\Omega(1)$ success probability, the cost of the naive application of each algorithm is $\widetilde{O}(\kappa^3)$. When the standard amplitude amplification [47] is used, one can reduce the number of repetitions from $\widetilde{O}(\kappa^2)$ to $\widetilde{O}(\kappa)$, and hence the complexity is reduced to $\widetilde{O}(\kappa^2)$. Furthermore, while the circuit depth of all methods above is independent of the right-hand side $|b\rangle$, the number of repetitions can depend on $|b\rangle$. Hence, the total number of queries can scale better than $\widetilde{O}(\kappa^2)$ (see Appendix B for detailed discussion in the context of QSVT).

In order to further reduce the worst-case complexity with respect to κ, it is possible to use a technique called variable-time amplitude amplification (VTAA) [1], which is a generalization of the standard amplitude amplification algorithm and allows to amplify the success probability of quantum algorithms by stopping different branches at different times. In [1], VTAA was first used to successfully improve the dependence of the HHL algorithm on κ to be almost linear, and the query complexity is $\widetilde{O}(\kappa/\epsilon^3)$. In [6], VTAA was combined with a low-precision phase estimate to improve the complexity of LCU to $\widetilde{O}(\kappa\,\mathrm{polylog}(1/\epsilon))$, which is near optimal with respect to both κ and ϵ. A similar strategy may be applied to accelerate QSVT. It is worth noting that the VTAA algorithm is a complicated procedure and can be difficult to implement.

Inspired by adiabatic quantum computation (AQC) [70–72], the recently developed randomization method (RM) [10] can solve QLSP with a runtime complexity $O(\kappa \ln(\kappa)/\epsilon)$, which is the first algorithm to yield $\widetilde{O}(\kappa)$ complexity without using VTAA. One can use an optimal Hamiltonian simulation

[2]The original HHL algorithm was not formulated in the language of block encoding. However, this does not affect the query complexity. We refer readers to section F for more details.

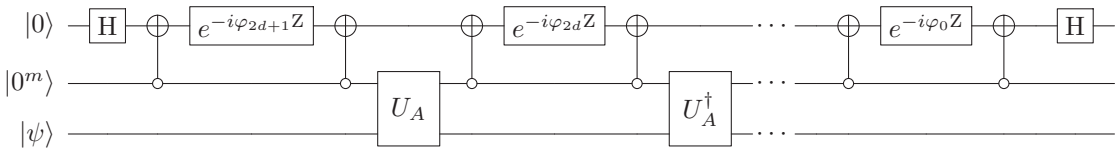


FIG. 4. Quantum circuit for implementing the quantum singular value transformation $f^{\diamond}(A/\alpha)$, where f is a real, odd polynomial of degree $2d+1$ satisfying Eq. (B4).

method (e.g., [37]) to yield a gate-based implementation of RM, of which the query complexity becomes $\widetilde{O}(\kappa/\epsilon)$. This is significantly better than the bounds proven using the vanilla AQC method, of which the complexity is $\widetilde{O}(\kappa^2/\epsilon)$ [10,71]. Using a fast eigenpath traversal method [73], which relies on repeated usage of phase estimation, the cost of RM can be further reduced to be near optimal as $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$.

Since repeated usage of phase estimation or VTAA are both difficult to implement and can require a significant number of ancilla qubits, it remains of great interest to obtain alternative algorithms to solve QLSP with near-optimal complexity scaling without resorting to such procedures. The first algorithm along this line is achieved by an optimally scheduled AQC algorithm [2], called AQC(exp), and the runtime complexity is $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$. There is also a slightly more versatile class of methods called AQC(p), of which the runtime complexity is simply $O(\kappa/\epsilon)$ [2]. Both AQC(exp) and AQC(p) require the implementation of a time-dependent simulation procedure. Using the recently developed near-optimal method for time-dependent Hamiltonian simulation [68], the query complexity of AQC(exp) is also $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$. This immediately implies that the optimal runtime complexity of the quantum approximate optimization algorithm (QAOA) [74] for solving QLSP is also $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$, as is verified by numerical experiments [2]. Using a different strategy called quantum eigenstate filtering [9], one can boost any algorithm that prepares a solution to $|x\rangle$ with $O(1)$ accuracy $O(\epsilon)$ with $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$ complexity. This is a very simple procedure and has a fully gate-based implementation via QSVT. In particular, using a method inspired by the quantum Zeno effect [9] obtains a simple algorithm to solve QLSP with $O(\kappa\,\mathrm{polylog}(\kappa/\epsilon))$ complexity, without using any amplitude amplification, phase estimation, or Hamiltonian simulation (time independent or time dependent). We remark that for both AQC(exp) and quantum eigenstate filtering, the result is a pure state, and the success probability of a single run is already $\Omega(1)$. This avoids the problem of repeated measurements inherent to HHL, LCU, and QSVT.

APPENDIX B: QUANTUM SINGULAR VALUE TRANSFORMATION, AND ITS APPLICATION TO QUANTUM LINEAR SYSTEM PROBLEM

For a square matrix $A \in \mathbb{C}^{N\times N}$, where for simplicity we assume $N = 2^n$ for some positive integer n, the singular value decomposition (SVD) of the normalized matrix A can be written as

$$A = W\Sigma V^{\dagger} \tag{B1}$$

or, equivalently,

$$A|v_k\rangle = \sigma_k|w_k\rangle, \quad A^{\dagger}|w_k\rangle = \sigma_k|v_k\rangle, \quad k \in [N]. \tag{B2}$$

We may apply a function $f(\cdot)$ on its singular values and define the generalized matrix function [75,76] as below.

Definition 4 (Generalized matrix function [75], Definition 4]). Let $f : \mathbb{R} \to \mathbb{R}$ be a scalar function such that $f(\sigma_i)$ is defined for all $i = 1, 2, \ldots, N$. The generalized matrix function is defined as

$$f^{\diamond}(A) := W f(\Sigma) V^{\dagger}, \tag{B3}$$

where

$$f(\Sigma) = \mathrm{diag}\,(f(\sigma_1), f(\sigma_2), \ldots, f(\sigma_N)).$$

This implies that the singular values satisfy

$$0 \leqslant \sigma_k \leqslant 1, \quad k \in [N].$$

In the discussion below, we assume f is a *real, odd polynomial of finite degree*, and satisfies

$$|f(x)| \leqslant 1, \quad \forall\, x \in [-1, 1]. \tag{B4}$$

We assume there is an $(\alpha, m, 0)$ block encoding of A denoted by U_A, so that the singular values of A/α are in [0,1]. The quantum singular value transformation (QSVT) [7, Theorem 2] provides an efficient way to implement $f^{\diamond}(A/\alpha)$ on a quantum computer using a very simple circuit, shown in Fig. 4. It only uses one extra qubit. Here, H and Z are the Hadamard and Pauli-Z gates, respectively. The polynomial degree is assumed to be $2d+1$ for $d \in \mathbb{N}$. The real numbers $\{\varphi_i\}_{i=0}^{2d+1}$ are called the phase factors of the QSVT circuit, and the block encoding U_A and its adjoint $U_A^{\dagger}$ appear alternatively. For a given polynomial f, the phase factors are an effective way to encode the polynomial in SU(2) [7, Corollary 11]. Although the computation of phase factors can be entirely carried out on classical computers, it is a nontrivial task to compute these phase factors. Significant progress has been achieved recently, enabling robust computation of phase factors for polynomials of degrees ranging from thousands to tens of thousands [77–79].

Remark 11. The condition on f in QSVT appears to be much stronger than that in Definition 4. Indeed, when f is a real, even polynomial, the counterpart of Fig. 4 implements a type of generalized matrix function taking a different form from Eq. (B3). Since the solution of QLSP only uses the formulation with odd polynomials, we refer interested readers to [7] for the construction of QSVT for real, even polynomials, as well as more general complex valued polynomials.

If f is a general odd function, we may first approximate $f(x)$ on the interval $[-1, 1]$ using a degree-d odd polynomial [we may apply a scaling factor if needed to satisfy Eq. (B4)], and apply QSVT to the resulting polynomial.

The matrix inversion can be implemented as a quantum singular value transformation in the following way: when A

is invertible $A^{-1} = V\Sigma^{-1}W^\dagger$. Therefore,

$$(A/\alpha)^{-1} = (f^\diamond(A/\alpha))^\dagger, \tag{B5}$$

where $f(x) = x^{-1}$. However, $f(\cdot)$ here is not bounded by 1 and is in fact singular at $x = 0$. Therefore, instead of approximating $f(x) = x^{-1}$ on the whole interval $[-1, 1]$ we consider an odd polynomial $p(x)$ such that

$$\left| p(x) - \frac{3\delta}{4x} \right| \leqslant \epsilon', \quad \forall\, x \in [-1, -\delta] \cup [\delta, 1]$$

and $|p(x)| \leqslant 1$ for all $x \in [-1, 1]$. The existence of such an odd polynomial of degree $O(\frac{1}{\delta}\ln(\frac{1}{\epsilon'}))$ is guaranteed by [45, Corollary 69].

Then [7, Theorem 2] enables us to implement $(p^\diamond(A/\alpha))^\dagger = V p(\Sigma/\alpha)W^\dagger$. We have

$$\begin{aligned} &\|(p^\diamond(A/\alpha))^\dagger - (3\delta/4)(A/\alpha)^{-1}\| \\ &\quad = \|p(\Sigma/\alpha) - (3\delta/4)(\Sigma/\alpha)^{-1}\| \leqslant \epsilon', \end{aligned} \tag{B6}$$

if all diagonal elements of Σ/α, i.e., the singular values of A/α, are in the interval $[\delta, 1]$. Therefore, we want all singular values of A/α to be at least δ distance away from the origin. We then use QSVT to block encode $(p^\diamond(A/\alpha))^\dagger$ given a block encoding of A.

We assume A can be accessed by its $(\alpha, m, 0)$ block encoding U_A. Let κ be the condition number of A, then the singular values of A/α are contained in $[\|A\|/(\alpha\kappa), \|A\|/\alpha]$. Therefore, we choose $\delta = \|A\|/(\alpha\kappa)$. Using QSVT, a $(1, m+1, 0)$ block encoding of $p^\diamond(A/\alpha)$ can be implemented [7, Theorem 2]. We denote this block encoding by $\mathcal{U}$. Then, by Eq. (B6)

$$\begin{aligned} &\left\| \frac{4}{3\delta\alpha}(\langle 0^{m+1} \otimes I|)\mathcal{U}^\dagger(|0^{m+1} \otimes I\rangle) - A^{-1} \right\| \\ &\quad = \left\| \frac{4}{3\delta\alpha}(p^\diamond(A/\alpha)^\dagger) - A^{-1} \right\| \leqslant \frac{4\epsilon'}{3\delta\alpha}. \end{aligned}$$

Therefore, $\mathcal{U}^\dagger$ is a $(4/(3\delta\alpha), m+1, 4\epsilon'/(3\delta\alpha))$ block encoding of A^{-1}. Because the cost of QSVT scales linearly with respect to the degree of the polynomial $p(x)$, the total number of queries to to U_A and its inverse is

$$O\left(\frac{1}{\delta}\ln\left(\frac{1}{\epsilon'}\right)\right) = O\left(\frac{\alpha\kappa}{\|A\|}\ln\left(\frac{1}{\epsilon'}\right)\right).$$

With the block encoding $\mathcal{U}^\dagger$ of matrix inversion we are then able to solve the linear system $A|x\rangle \propto |b\rangle$ when we are given the quantum state $|b\rangle$ representing the right-hand side of the QLSP. We assume $|b\rangle$ can be accessed through the oracle U_b such that

$$U_b|0^n\rangle = |b\rangle.$$

We introduce the parameter

$$\xi = \|A^{-1}|b\rangle\|,$$

which plays an important part in the success probability of the procedure. This parameter also appears in Secs. II A, II B, and Corollary 1. Let $|\widetilde{x}\rangle = (3\delta/4)(A/\alpha)^{-1}|b\rangle$, then the normalized solution state is $|x\rangle = |\widetilde{x}\rangle/\||\widetilde{x}\rangle\|$ satisfying $A|x\rangle \propto |b\rangle$. Now denote $|\widetilde{y}\rangle = (p^\diamond(A/\alpha))^\dagger|b\rangle$, and $|y\rangle = |\widetilde{y}\rangle/\||\widetilde{y}\rangle\|$. If the polynomial approximation has error ϵ', then we have $\||\widetilde{x}\rangle - |\widetilde{y}\rangle\| \leqslant \epsilon'$. However, for the normalized state $|y\rangle$, this error is scaled accordingly. When $\epsilon' \ll \||\widetilde{x}\rangle\|$, we have

$$\||x\rangle - |y\rangle\| \approx \frac{\||\widetilde{x}\rangle - |\widetilde{y}\rangle\|}{\||\widetilde{x}\rangle\|} \leqslant \frac{\epsilon'}{\||\widetilde{x}\rangle\|}. \tag{B7}$$

Also, we have

$$\||\widetilde{x}\rangle\| = \left\| \frac{3\delta}{4}\left(\frac{A}{\alpha}\right)^{-1}|b\rangle \right\| = \frac{3\alpha\delta\xi}{4} \geqslant \frac{3\|A\|\xi}{4\kappa}.$$

Therefore, in order for the normalized output quantum state to be ϵ close to the normalized solution state $|x\rangle$, we need to set $\epsilon' = O(\epsilon\|A\|\xi/\kappa)$.

The success probability of the above procedure is $\Omega(\||\widetilde{x}\rangle\|^2) = \Omega(\|A\|^2\xi^2/\kappa^2)$. With amplitude amplification we can boost the success probability to be greater than $\frac{1}{2}$ with one qubit serving as a witness, i.e., if measuring this qubit we get an outcome 0 it means the procedure has succeeded, and if 1 it means the procedure has failed. It takes $O(\kappa/\|A\|\xi)$ rounds of amplitude amplification, i.e., using $\mathcal{U}^\dagger$, $\mathcal{U}$, U_b, and $U_b^\dagger$ for $O(\kappa/\|A\|\xi)$ times. A single $\mathcal{U}$ or its inverse uses U_A and its inverse

$$O\left(\frac{1}{\delta}\ln\left(\frac{1}{\epsilon'}\right)\right) = O\left(\frac{\alpha\kappa}{\|A\|}\ln\left(\frac{\kappa}{\epsilon\|A\|\xi}\right)\right)$$

times. Therefore, the total number of queries to U_A and its inverse is

$$O\left(\frac{\kappa}{\|A\|\xi} \times \frac{\alpha\kappa}{\|A\|}\ln\left(\frac{\kappa}{\epsilon\|A\|\xi}\right)\right) = O\left(\frac{\alpha\kappa^2}{\|A\|^2\xi}\ln\left(\frac{\kappa}{\|A\|\xi\epsilon}\right)\right).$$

The number of queries to the U_b and its inverse is $O(\kappa/\|A\|\xi)$. To summarize, we have the refined version of using QSVT to solve QLSP:

Theorem 7 (Standard QSVT linear system solver). Let U_A be an $(\alpha, m, 0)$ block encoding of A with condition number κ, U_b be the oracle preparing the right-hand side vector $|b\rangle$, and $\xi = \|A^{-1}|b\rangle\| \in [1/\|A\|, \|A^{-1}\|]$. Then the solution $|x\rangle \propto A^{-1}|b\rangle$ can be obtained with precision ϵ and with a success probability at least $\frac{1}{2}$, using $O(\alpha\kappa^2/[\|A\|^2\xi)\ln(\kappa/(\|A\|\xi\epsilon)])$ queries to U_A and $U_A^\dagger$, and $O(\kappa/(\|A\|\xi))$ queries to U_b.

We consider the following two cases for the magnitude of ξ. For simplicity we assume $\alpha = \Theta(\|A\|)$.

(1) In general if no further promise is given, then $\xi \geqslant 1/\|A\|$. The total query complexity of U_A is therefore $O(\kappa^2\ln(\kappa/\epsilon))$. This is the typical complexity referred to in the literature.

(2) If $|b\rangle$ has a $\Omega(1)$ overlap with the left-singular vector of A with the smallest singular value, then $\xi = \Omega(\|A^{-1}\|) = \Omega(\kappa/\|A\|)$. This is the best-case scenario, and the total query complexity of U_A is $O(\kappa\ \ln(1/\epsilon))$, and the number of queries to the right-hand side vector $|b\rangle$ is $O(1)$, which is independent of the condition number.

APPENDIX C: FAST INVERSION OF 1-SPARSE MATRICES

In this Appendix we discuss the fast inversion of 1-sparse matrices, i.e., there is at most one nonzero matrix element in every row and column of the matrix. First, let us assume that there exists an efficient function f such that $f(x)$ yields the column number for the nonzero matrix element of A in row x.

We then define the oracle $O_f : |x\rangle|c\rangle \mapsto |x\rangle|c \oplus f(x)\rangle$ as well as the oracle $O_A : |x\rangle|y\rangle|c\rangle \mapsto |x\rangle|y\rangle|c \oplus A_{xy}\rangle$. Without loss of generality we assume A is a Hermitian matrix [otherwise we can dilate it into a Hermitian matrix as in Eq. (5); in this case we also need access to the oracle $O_g : |x\rangle|c\rangle \mapsto |x\rangle|c \oplus g(x)\rangle$, where $g(x)$ yields the row number for the nonzero matrix element of A in column x]. For convenience, we will assume that the matrix elements of A are encoded in polar form.

Lemma 6. Let A be a 1-sparse Hermitian, invertible, and row-computable matrix in $\mathbb{C}^{2^n \times 2^n}$ where each A_{xy} can be exactly represented using n' bits of precision then for any $\delta > 0$, an $((\min_x |A_{xf(x)}|)^{-1}, 1, \delta)$ block encoding of A^{-1} can be implemented using at most 4 oracle queries and an $O(\text{poly}(n' \ln(1/\delta)))$ auxiliary operations taken from a gate set consisting of H, T,CNOT.

Proof. The proof follows standard reasoning provided in the quantum simulation literature [28,29]. The first observation that is important in the reasoning is that all Hermitian 1-sparse matrices can be written as a direct sum of irreducible one- and two- dimensional matrices. This direct sum structure can be exploited to allow the matrix inverse to be explicitly computed on each of these subspaces of constant dimension and thereby the desired transformation can be performed to invert the matrix.

Consider a fixed basis vector as the input denoted by $|x\rangle$. We need to consider two cases. First, let $|x\rangle$ be part of an irreducible two-dimensional block of A. This means that $f(x) \neq x$. It then follows that

$$\begin{aligned} A[a|x\rangle + b|f(x)\rangle] &= aA_{f(x)x}|f(x)\rangle + bA_{xf(x)}|x\rangle \\ &= |A_{xf(x)}|\left(\frac{aA^*_{xf(x)}}{|A_{xf(x)}|}|f(x)\rangle + \frac{bA_{xf(x)}}{|A_{xf(x)}|}|x\rangle\right). \end{aligned} \tag{C1}$$

This justifies the claim that $|x\rangle$ is in an irreducible two-dimensional space. Additionally, we can see by applying A again to this result and using that A is Hermitian, the eigenvalues of A within the two-dimensional subspace are $\pm|A_{xf(x)}|$. Similarly, the eigenvectors can be taken to be

$$|\psi_x^{\pm}\rangle := \frac{1}{\sqrt{2}}\left(|\min(x, f(x))\rangle \pm \frac{A^*_{xf(x)}}{A_{xf(x)}}|\max(x, f(x))\rangle\right), \tag{C2}$$

where the eigenvectors corresponding to the positive and negative eigenvalues are $|\psi_x^+\rangle$ and $|\psi_x^-\rangle$, respectively.

In order to perform this eigendecomposition of the input vectors, we will map this two-dimensional space to a single-qubit space and then diagonalize the operator once it has been reduced to this case. There are several operations that we need to define for this diagonalization process. First, let us define CMP to be a comparator circuit. This self-inverse unitary transformation has the action (for any $b \in \mathbb{Z}_2$)

$$\text{CMP} : |x\rangle|y\rangle|b\rangle \mapsto \begin{cases} |x\rangle|y\rangle|b\rangle, & \text{if } x \leqslant y \\ |x\rangle|y\rangle|b \oplus 1\rangle, & \text{if } x > y. \end{cases} \tag{C3}$$

The CMP circuit can be implemented using a two-complement adder [80] using a polynomial number of gates in the bits of precision of the inputs x and y. The next operation we need is the controlled swap gate CSWAP which we define such that for any two states $|\psi\rangle$ and $|\phi\rangle$ of the same dimension

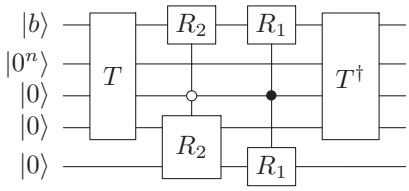


FIG. 5. Schematic quantum circuit for implementing the one- and two-dimensional transformations. R_1 and R_2 refer to the rotations implemented on the one- and two-dimensional subspaces, respectively.

with $\text{SWAP}|\psi\rangle|\phi\rangle = |\phi\rangle|\psi\rangle$

$$\text{CSWAP} = |1\rangle\langle 1| \otimes \text{SWAP} + |0\rangle\langle 0| \otimes I. \tag{C4}$$

The CSWAP operation can be implemented using a number of CNOT and Toffoli gates that is linear in the bits of $|\psi\rangle$ and $|\phi\rangle$. We denote this transformation T and implement it through the following steps under the assumption (without loss of generality) that $x \leqslant f(x)$:

$$\begin{aligned} &(a|x\rangle + b|f(x)\rangle)|0^{n+1}\rangle \\ &\quad \mapsto_{O_f} \quad (a|x, f(x)\rangle + b|f(x), x\rangle)|0^{n'+2}\rangle \\ &\quad \mapsto_{\text{CMP}} \quad (a|x, f(x)\rangle|0\rangle + b|f(x), x\rangle|1\rangle)|0^{n'+1}\rangle \\ &\quad \mapsto_{\text{CSWAP}} \quad |x, f(x)\rangle(a|0\rangle + b|1\rangle)|0^{n'+1}\rangle \\ &\quad \mapsto_{O_A} \quad |x, f(x)\rangle(a|0\rangle + b|1\rangle)|0\rangle|A_{xf(x)}\rangle \\ &\quad \mapsto_{\text{CNOT}} \quad |x, f(x) \oplus x\rangle(a|0\rangle + b|1\rangle)|0\rangle|A_{xf(x)}\rangle \\ &\quad \mapsto_{\Lambda_n(\text{NOT})} \quad |x, f(x) \oplus x\rangle(a|0\rangle + b|1\rangle)|\delta_{x,f(x)}\rangle|A_{xf(x)}\rangle. \end{aligned} \tag{C5}$$

Here, $\Lambda_n(\text{NOT})$ is an n-controlled not gate. This sequence of operations mirrors standard approaches for simulating 1-sparse Hamiltonian dynamics as given in [30,81].

All the information needed to implement the inverse of A on each one- or two-dimensional irreducible subspace of the 1-sparse matrix is computed using a single invocation of T. From that we invert the matrix using two controlled inversion circuits R_1 and R_2 which correspond to the one- and two-dimensional inversion circuit, respectively. This strategy is shown schematically in Fig. 5.

The transformation R_2 in Fig. 5 can be implemented as follows. If $|x\rangle$ and $|f(x)\rangle$ form an irreducible two-dimensional subspace, then we can identify R_2 through the following reasoning. First, note that

$$\begin{aligned} &\frac{A}{|A_{xf(x)}|} \otimes I(a|x\rangle + b|f(x)\rangle)|0^{n+n'+2}\rangle \\ &\quad = T^\dagger(I \otimes R_z(\text{Arg}(A_{xf(x)}))XR_z(-\text{Arg}(A_{xf(x)})) \otimes I) \\ &\qquad \times T(a|x\rangle + b|f(x)\rangle)|0^{n'+n+2}\rangle, \end{aligned} \tag{C6}$$

here X is the Pauli-X gate. Next, we see from the fact that if C is a 2×2 invertible Hermitian matrix, then

$$C = \begin{bmatrix} 0 & c \\ c^* & 0 \end{bmatrix} \quad \Rightarrow \quad C^{-1} = \begin{bmatrix} 0 & (c^{-1})^* \\ c^{-1} & 0 \end{bmatrix} \tag{C7}$$

that

$$\frac{A^{-1}\otimes I}{|A_{xf(x)}|^{-1}}(a|x\rangle+b|f(x)\rangle)|0^{n+n'+2}\rangle$$
$$=P_2^{\dagger}\big(I\otimes R_z\big(\mathrm{Arg}\big(A^*_{xf(x)}\big)\big)XR_z\big(-\mathrm{Arg}\big(A^*_{xf(x)}\big)\big)\otimes I\big)$$
$$\times\ P_2(a|x\rangle+b|f(x)\rangle)|0^{n'+n+2}\rangle. \tag{C8}$$

Equation (C8) shows how we can block encode the normalized inverse; however, since the value of $|A_{xf(x)}|$ need not be constant over x, we will need to show how to construct a block encoding that will allow the normalization factor to be varied across each block. The central observation is that

$$A^{-1}\otimes I(a|x\rangle+b|f(x)\rangle)|0^{n+n'+2}\rangle$$
$$=(I\otimes I\otimes\langle 0|)\left(\frac{A^{-1}\otimes I}{|A_{xf(x)}|^{-1}}(a|x\rangle+b|f(x)\rangle)|0^{n+n'+2}\rangle\right.$$
$$\left.\otimes\left(\frac{\min_x|A_{xf(x)}|}{|A_{xf(x)}|}|0\rangle+\sqrt{1-\left(\frac{\min_x|A_{xf(x)}|}{|A_{xf(x)}|}\right)^2}|1\rangle\right)\right). \tag{C9}$$

Then, by applying the procedure described in (C8) and (C9) we can therefore block encode A^{-1}. We call this operation R_2 in Fig. 5. Since R_2 is unitary, it is also linear and therefore will apply the inverse on every irreducible two-dimensional subspace simultaneously.

This procedure requires a circuit of polynomial size in the bits of precision n' to compute the inverse trigonometric function and reciprocal needed to perform this rotation. However, this requires no queries and is efficient given n' and therefore the precise details of the arithmetic circuits used are irrelevant for our lemma. The one aspect that is relevant is that even if the inputs of A_{xy} require finite precision, the computation of the arccosine will often require infinite precision to precisely represent. This cost is $O(\mathrm{poly}(n'\ln(1/\delta)))$ quantum operations. Thus, we can perform an $(|A_{xy}|^{-1},1,\delta)$ block encoding of the inverse assuming each irreducible subspace is two dimensional.

The construction for R_1 in Fig. 5 is even simpler than that for R_2. In this case $|x\rangle$ is a member of an irreducible one-dimensional subspace. This occurs when $f(x)=x$ or, in other words, when the nonzero matrix element in row x of A occurs on the diagonal. In this case,

$$A^{-1}|x\rangle=\frac{1}{A_{xx}}|x\rangle=\frac{\mathrm{sgn}(A_{xx})}{|A_{xx}|}|x\rangle. \tag{C10}$$

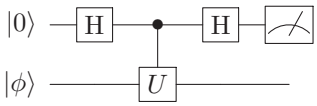
Note that because A is by assumption Hermitian, $A_{xx}\in\mathbb{R}$. This further means that the diagonalization steps that were needed for the two-dimensional case are unnecessary here. Thus, in this case we can express the inverse as

$$|x\rangle|0\rangle\mapsto\mathrm{sgn}(A_{xx})|x\rangle\left(\frac{\min_x|A_{xf(x)}|}{|A_{xx}|}|0\rangle\right.$$
$$\left.+\sqrt{1-\left(\frac{\min_x|A_{xf(x)}|}{|A_{xx}|}\right)^2}|1\rangle\right). \tag{C11}$$

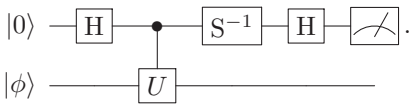
Thus, the inversion process in this case looks very similar to the two-dimensional case after applying the transformation T. Then, by replacing the Pauli-X operation in (C8) with a controlled-Z operation with the output of this function as control, we can selectively diagonalize the block. Similarly, if $x=f(x)$ we need to flip the sign; this can be achieved by comparing A_{xx} to zero applying Z to the resulting qubit that stores whether $A_{xx}<0$ with the bit that stores $|(f(x)\neq x)\rangle$. This can also be achieved using an $O(n)$-size circuit consisting of NOT, CNOT, and Toffoli. The part of the circuit that computes θ_{xy} is common to both cases and thus does not need to be changed. Thus, if we augment the circuit by making these changes, we can implement a $((\min_{x,y}|A_{xy}|)^{-1},1,\delta)$ block encoding of the matrix inverse regardless of whether $|x\rangle$ is in an irreducible one- or two-dimensional subspace using 4 oracle queries and $O(\mathrm{poly}((n+n')\ln(1/\delta)))$ auxiliary gate operations from the R_z, H, CNOT, Toffoli gate library, which can be compiled down at polynomial cost to gates taken from H, T, CNOT. This completes the proof. ■

APPENDIX D: HADAMARD TEST FOR NONUNITARY MATRICES

The well-known Hadamard test is frequently used to obtain the expectation value of a unitary operator. Suppose we want to obtain $\langle\phi|U|\phi\rangle$ for some unitary U and $|\phi\rangle$, then for the real part we need the following circuit:

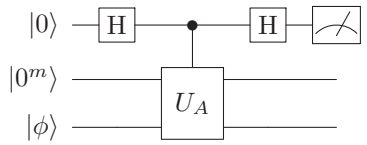


and for the imaginary part we need the circuit that is slightly different



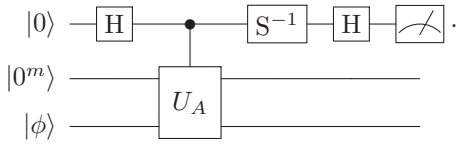
Here H is the Hadamard gate and S is the phase gate. We find that the probabilities of obtaining 0 when measuring the first qubit are $\frac{1}{2}(1+\mathrm{Re}\,\langle\phi|U|\phi\rangle)$ and $\frac{1}{2}(1+\mathrm{Im}\,\langle\phi|U|\phi\rangle)$, respectively, for the two circuits.

A small modification gives us a way of computing expectation value of nonunitary matrices when given the block encodings. If we have the $(\alpha,m,0)$ block encoding of A which we denote by U_A, then for the real part of $\langle\phi|A|\phi\rangle$ we consider the following circuit:



(D1)

and for the imaginary part we consider



(D2)

The probabilities of obtaining 0 when measuring the first qubit are $\frac{1}{2}(1+\frac{1}{\alpha}\,\mathrm{Re}\,\langle\phi|A|\phi\rangle)$ and $\frac{1}{2}(1+\frac{1}{\alpha}\,\mathrm{Im}\,\langle\phi|A|\phi\rangle)$, respectively. One can derive these two probabilities easily by noting the fact that $(\langle 0^m|\langle\phi|)U_A(|0^m\rangle|\phi\rangle)=\frac{1}{\alpha}\langle\phi|A|\phi\rangle$.

It will be straightforward to estimate the probability of obtaining all 0's in measurement through Monte Carlo sampling, and thereby estimate the quantity of interest $\langle\phi|A|\phi\rangle$. The efficiency of the Monte Carlo sampling can be generally accelerated by the amplitude estimation procedure [47, Theorem 12]. However, there are two issues we need to take into account, both arising from the preparation of $|\phi\rangle$:

(1) The algorithm for preparing $|\phi\rangle$ may only be able to prepare a state close to it, which we denote as $|\widetilde{\phi}\rangle$ with trace distance $\sqrt{1-|\langle\widetilde{\phi}|\phi\rangle|^2}\leqslant\varsigma$.

(2) The algorithm for preparing $|\phi\rangle$ may involve measurement and have success probability lower bounded by $p<1$.

The first issue compels us to take the error of the state preparation into account. The second issue requires some further explanation. If we wanted to estimate $\langle\phi|A|\phi\rangle$ using Monte Carlo sampling, then we could simply measure the ancilla qubit for each successful preparation of the state $|\phi\rangle$, and do nothing when the state preparation is unsuccessful. However, when we want to use amplitude estimation, it is no longer possible to directly select only the successful instances in this way. In order to construct the reflection operator needed in the amplitude amplification, we will need to prepare the state $|\phi\rangle$, or a state close to it, using a unitary circuit with success probability 1.

First, we assume $|\phi\rangle$ can be prepared perfectly with probability 1 using a unitary circuit U_ϕ. When estimating the real part using amplitude estimation, we will need to run U_ϕ, the circuit in (D1), and their inverses $O(\alpha/\epsilon)$ times to estimate the real part to precision ϵ. The same is true for the imaginary part.

Next, we consider when the preparation involves some error. Due to the trace distance bound of $|\widetilde{\phi}\rangle$, we have

$$|\langle\phi|A|\phi\rangle-\langle\widetilde{\phi}|A|\widetilde{\phi}\rangle|\leqslant 2\|A\|\varsigma.$$

Finally, we consider the case when $|\widetilde{\phi}\rangle$ is produced only with probability at least p. Formally, we assume $(\langle 0^r|\otimes I_n)U_\phi|0^r\rangle|0^n\rangle=a|\widetilde{\phi}\rangle$ where $a\geqslant\sqrt{p}$. In this case we first apply the fixed-point amplitude amplification [45, Theorem 27] to U_ϕ. Compared to the standard amplitude amplification [47], the fixed-point amplitude amplification has the advantage of using only a unitary circuit, and can boost the success probability 1. By [45, Theorem 27], there exists a unitary circuit $\widetilde{U}_\phi$ such that $\||0^r\rangle|\widetilde{\phi}\rangle-\widetilde{U}_\phi|0^r\rangle|0^n\rangle\|\leqslant\epsilon'$, and this circuit uses U_ϕ and its inverse $O(\frac{1}{\sqrt{p}}\ln(\frac{1}{\epsilon'}))$ times. We may set we $\epsilon'=O(\varsigma)$, so that the trace distance of the output away from $|0^r\rangle|\widetilde{\phi}\rangle$ is $O(\varsigma)$.

A problem with amplitude estimation is that there is a failure probability, i.e., the estimated amplitude differs from the true amplitude by more than the allowed error ϵ, and this failure probability as mentioned in [47, Theorem 12] is at most $1-8/\pi^2<\frac{1}{2}$. If we want the failure probability to be smaller than δ, then we can run the amplitude estimation multiple times and take the median. Using the Chernoff-Hoeffding theorem we can estimate that we only need to run $O(\ln(1/\delta))$ times to ensure the failure probability, i.e., the probability that the median differs from the true amplitude by more than ϵ, is at most δ. Therefore, we have the following lemma:

Lemma 7. Suppose a state $|\phi\rangle$ can be prepared with trace-distance error ς by a unitary circuit U_ϕ with probability at least p, and A is given through its $(\alpha,m,0)$ block encoding U_A, then $\langle\phi|A|\phi\rangle$ can be estimated using amplitude estimation to precision $2\alpha\varsigma+\epsilon$ with probability at least $1-\delta$, using $O((\alpha/\epsilon)\ln(1/\delta))$ applications of U_A and its inverse $O((\alpha/\sqrt{p}\epsilon)\ln(1/\varsigma)\ln(1/\delta))$ applications of U_ϕ and its inverse, and $O((\alpha/\sqrt{p}\epsilon)\ln(1/\varsigma)\ln(1/\delta))$ other one- and two-qubit gates.

APPENDIX E: COMPUTING IMAGINARY PARTS OF THE GREEN'S FUNCTION WITHOUT USING THE HADAMARD TEST

We remark that if we are interested in computing the imaginary part of the Green's function (more accurately, the anti-Hermitian part of the Green's function), the calculation can be simplified as follows. Let $z=E-i\eta$, and define

$$\Gamma^{(\pm)}(z)=\frac{1}{2i}[G^{(\pm)}(z)-(G^{(\pm)}(z))^\dagger],$$

which are real symmetric matrices. Consider $\Gamma^{(+)}$ for simplicity, then

$$\begin{aligned}\Gamma^{(+)}_{ij}(z)&=\langle\Psi_0^N|\hat{a}_i\,\mathrm{Im}(z-[\hat{H}-E_0])^{-1}\hat{a}_j^\dagger|\Psi_0^N\rangle\\&=\eta\langle\Psi_0^N|\hat{a}_i((E\quad+E_0-\hat{H})^2+\eta^2)^{-1}\hat{a}_j^\dagger|\Psi_0^N\rangle.\end{aligned}$$

Note that the diagonal entries are

$$\begin{aligned}\Gamma^{(+)}_{ii}(z)&=\eta\langle\Psi_0^N|\hat{a}_i((E+E_0-\hat{H})^2+\eta^2)^{-1}\hat{a}_i^\dagger|\Psi_0^N\rangle\\&=\eta\|(z+E_0-\hat{H})^{-1}\hat{a}_i^\dagger|\Psi_0^N\rangle\|_2^2.\end{aligned}$$

If we solve the QLSP

$$(z+E_0-\hat{H})|y_i\rangle=\hat{a}_i^\dagger|\Psi_0^N\rangle,$$

the success probability in measuring the ancilla qubits and obtain all 0's is

$$\left(\alpha'_{z+E_0-\hat{H}}\right)^{-2}\|(z+E_0-\hat{H})^{-1}\hat{a}_i^\dagger|\Psi_0^N\rangle\|_2^2.$$

Here, $\alpha'_{z+E_0-\hat{H}}$ is the subnormalization factor for $(z+E_0-\hat{H})^{-1}$. Hence, $\Gamma^{(+)}_{ii}(z)$ can be directly computed from the success probability without using the Hadamard test.

In order to obtain the off-diagonal entries, we simply use the identity

$$\begin{aligned}\Gamma^{(+)}_{ij}(z)=\frac{\eta}{2}\big(&\langle\Psi_0^N|(\hat{a}_i+\hat{a}_j)((E+E_0-\hat{H})^2+\eta^2)^{-1}(\hat{a}_i^\dagger+\hat{a}_j^\dagger)|\Psi_0^N\rangle\\&-\langle\Psi_0^N|\hat{a}_i((E+E_0-\hat{H})^2+\eta^2)^{-1}\hat{a}_i^\dagger|\Psi_0^N\rangle\\&-\langle\Psi_0^N|\hat{a}_j((E+E_0-\hat{H})^2+\eta^2)^{-1}\hat{a}_j^\dagger|\Psi_0^N\rangle\big),\end{aligned}$$

which can again be evaluated as success probabilities for separately solving three linear systems. The treatment of $\Gamma^{(-)}$ follows the same procedure.

APPENDIX F: QUERY COMPLEXITIES OF ESTIMATING GREEN'S FUNCTION USING HHL AND LCU/QSVT

We assume that we are given an exact block encoding of $\hat{H}$ denoted by U_H with subnormalization factor α_H. Then, supposing we know the ground energy E_0, we can construct

a block encoding of $z - \hat{H} + E_0$ with subnormalization factor $O(|z| + \alpha_H)$ using [7, Lemma 29].

To our best knowledge, the HHL algorithm has not been formulated in terms of block encoding in the literature, so for completeness we provide such a formulation below. The HHL algorithm relies on time evolution to solve the linear system. We assume the time evolution, with a Hamiltonian H to be specified, has a cost $O(\|H\|t)$, which is typical in many Hamiltonian simulation methods [30,36,37,40,68,82], and omit the dependence on the desired precision, because the dependence is only polylogarithmic and therefore does not play an important role in the complexity. The circuit construction of HHL effectively gives a block encoding of the matrix inverse. We explain it in detail below.

Assume we have a matrix M, given in its block encoding U_M with subnormalization factor α_M. If M is not Hermitian, we may consider the dilated matrix denoted by $\widetilde{M}$ as in Eq. (5). If we are able to block encode $\widetilde{M}^{-1}$, then extracting the lower-left block of this matrix through single-qubit operations will yield us M^{-1}. We then discuss how to get $\widetilde{M}^{-1}$ using HHL.

The HHL algorithm consists of the following steps: We start with two ancilla registers and a main register, containing r and one qubit, respectively, each ancilla qubit in the $|0\rangle$ state, and the main register starting with some state $|b\rangle$. First we perform Hadamard transform on the first ancilla register, which is usually called the clock register. Then we apply $\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau| \otimes e^{i\widetilde{M}\tau t_0/T}$, the controlled time evolution of $\widetilde{M}$, on the main register, controlled by the first ancilla register. Next we apply QFT on the first ancilla register, so that this register stores the approximate eigenvalues of $\widetilde{M}$ in superposition. Then we apply rotation on the second ancilla register, which contains only one qubit, controlled by the first register. Finally, we uncompute the first ancilla register, and measure the second ancilla register. Upon obtaining outcome 1 we have successfully prepared $\widetilde{M}^{-1}|b\rangle/\|\widetilde{M}^{-1}|b\rangle\|$.

The main source of error in HHL is the phase estimation step. To control the phase estimation error to be within δ we need to let $t_0 = O(1/\delta)$. However, when the eigenvalue λ is off by δ its inverse $1/\lambda$ will be off by approximately $2\delta/\lambda^2$ for $\delta \ll \lambda$. We denote the smallest singular value of M as $\sigma_{\min}$. Therefore, in order to have a block-encoding error of $\widetilde{M}$ to be smaller than ϵ', we need to let $t_0 = O(1/\epsilon'\sigma_{\min}^2)$. In the original HHL algorithm this dependence is mitigated because there is a subnormalization of the output quantum state involved. If we want M^{-1} to have a block-encoding error upper bounded by ϵ, we only need $\widetilde{M}^{-1}$ to have a block-encoding error upper bounded by $\alpha_M\epsilon$. Therefore, we can set $\epsilon' = \alpha_M\epsilon$.

In the case of $M = z - \hat{H} + E_0$, we have $\alpha_M = O(|z| + \alpha_H)$, the smallest singular value of M is lower bounded by η, and therefore $\sigma_{\min} = \Omega(\eta/\alpha_M)$. Therefore, we have $t_0 = O(\alpha_M/\eta^2\epsilon)$. The dominant cost of constructing block encoding of M^{-1} is running Hamiltonian simulation of $\widetilde{M}$ for time t_0. Therefore, a single block encoding of M^{-1} takes $O(\alpha_M/\eta^2\epsilon)$ applications of U_M and its inverse. The block encoding subnormalization factor is upper bounded by the block-encoding subnormalization of $\widetilde{M}^{-1}$, which is at most $1/\sigma_{\min}$, divided by α_M. It is therefore $O(1/\sigma_{\min}\alpha_M) = O(1/\eta)$. When we take the estimates for the number of queries to U_M and the subnormalization factor into Lemma 7, we arrive at the query complexity estimates in the first row of Table I.

The complexities of LCU and QSVT are both easier to estimate than HHL because of the polylogarithmic dependence on the desired block-encoding error, as both methods use results from approximation theory to make this possible. In both cases we first construct the block encoding of $\widetilde{M}^{-1}$, whose singular values are in $[1/\sigma_{\min}, 1]$, where $\sigma_{\min}$ is the same as defined before. To construct a block encoding of $\widetilde{M}^{-1}$ with error upper bounded by ϵ', $\widetilde{O}((1/\sigma_{\min})\text{polylog}(1/\epsilon'))$ queries to $U_{\widetilde{M}}$ are required. When we regard the resulting circuit as a block encoding of M^{-1} we need to set $\epsilon' = \alpha_M\epsilon$ as before. Therefore, in the application to Green's function a single block-encoding query U_M and its inverse $\widetilde{O}((\alpha_M/\eta)\text{polylog}(1/\epsilon))$ times. The resulting block-encoding subnormalization factor of M^{-1} scales the same as in HHL, which in the application to Green's function is upper bounded by $O(1/\eta)$. These estimates lead to the complexities in the second row of Table I.

APPENDIX G: PROOF OF LEMMA 2: DISCRETIZING THE CONTOUR INTEGRAL

In this Appendix we prove Lemma 2. The goal is to discretize the contour integral

$$I = \frac{1}{2\pi i}\oint_\Gamma \frac{e^{-\beta z}}{x-z}dz = \frac{-1}{2\pi i}\int_{-\infty}^{\infty} \frac{e^{-\beta(t^2-\zeta+it)}(2t+i)}{x-t^2+\zeta-it}dt, \tag{G1}$$

for $x > 0$ where Γ is defined in Eq. (34). We first want to approximate the integral on the real axis by an integral on a finite interval. We define

$$I_T = \frac{-1}{2\pi i}\int_{-T}^{T} \frac{e^{-\beta(t^2-\zeta+it)}(2t+i)}{x-t^2+\zeta-it}dt \tag{G2}$$

and

$$g(t,x) = \frac{e^{-\beta(t^2-\zeta+it)}(2t+i)}{x-t^2+\zeta-it}. \tag{G3}$$

Then we have, for $|t| \geqslant \frac{1}{2}$, $|g(t,x)| \leqslant \sqrt{8}e^{-\beta(t^2-\zeta)}$. This implies

$$|I - I_T| \leqslant \frac{\sqrt{2}e^{\beta\zeta}}{\pi}\int_{|t|>T} e^{-\beta t^2}dt = \sqrt{\frac{2}{\beta\pi}}e^{\beta\zeta}\text{erfc}(\sqrt{\beta}T),$$

when $T \geqslant \frac{1}{2}$. Due to the inequality [83]

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt \leqslant e^{-x^2},$$

we can bound the error due to the finite-range truncation as

$$|I - I_T| \leqslant \sqrt{\frac{2}{\beta\pi}}e^{\beta\zeta}e^{-\beta T^2}. \tag{G4}$$

It now remains to find a quadrature for I_T. To do this we use the Gauss-Legendre quadrature. We write out the Chebyshev expansion of $g(t,x)$ as

$$g(t,x) = \sum_{n=0}^{\infty}\hat{g}_n(x)T_n(t/T), \tag{G5}$$

where $T_n(t)$ is the nth Chebyshev polynomial of the first kind. We define

$$I_{\mathrm{GL}} = T \sum_{j\in[J]} w_j g(s_j T, x), \tag{G6}$$

where w_j and s_j are selected according to the standard Gauss-Legendre quadrature formula on $[-1, 1]$. Because this quadrature formula is exact for polynomials of degree up to $2J-1$, and that

$$\sum_{j\in[J]} w_j |T_n(s_j)| \leqslant 2,$$

we have

$$|I_T - I_{\mathrm{GL}}| \leqslant 2T \sum_{n\geqslant 2J} |\hat{g}_n(x)|. \tag{G7}$$

Therefore, we only need to bound the coefficients $|\hat{g}_n(x)|$.

Let $h(\theta, x) = g(T\cos(\theta), x)$, and write out its Fourier expansion

$$h(\theta, x) = \sum_{n=-\infty}^{\infty} \hat{h}_n(x) e^{in\theta},$$

then

$$\hat{g}_n(x) = \hat{h}_n(x) + \hat{h}_{-n}(x),\ n > 0, \quad \hat{g}_0(x) = \hat{h}_0(x).$$

Note that if we introduce a new variable $z = e^{i\theta}$, then

$$\widetilde{h}(z, x) = h(\theta, x) = \sum_{n=-\infty}^{\infty} \hat{h}_n(x) z^n,$$

which takes the form of a Laurent series. Therefore, the coefficients of the Chebyshev expansion $\{\hat{g}_n(x)\}_{n=0}^{\infty}$ are directly related to the coefficients of the Laurent expansion $\{\hat{h}_n(x)\}_{n=-\infty}^{\infty}$.

The function $g(t, x)$ is analytic in t in the domain $\{t \in \mathbb{C} : t^2 - \zeta + it - x \neq 0\}$. We need to estimate how far from the real axis $h(\theta, x)$ can be extended as an analytic function of θ in order to estimate the convergence rate of the Fourier series (see, e.g., [84, Chapter 8]).

We want to lower bound $|t^2 + it - x - \zeta|$ when $|\operatorname{Im} t|$ is bounded by some parameter b to be chosen. We write $t = w + iy$, and require $|y| \leqslant b$, then choose

$$\zeta = 2b(1-b). \tag{G8}$$

ζ is chosen in this way so that the distance between the positive part of the real axis and the set $\{t^2 + it - \zeta : |\operatorname{Im} t| \leqslant b\}$ can be bounded from below, which we will consider next. For each fixed $\operatorname{Im} t = y$, the image of $\{t^2 + it - \zeta : \operatorname{Im} t = y\}$ is a parabola in the complex plane parametrized by w. The parabola moves to the left and widens when y increases from $-b$ to b (see Fig. 6 for an illustration). Therefore, we only need to consider when $y = -b$. In this case

$$|t^2 + it - x - \zeta|^2 = [w^2 - b(1-b) - x]^2 + w^2(1-2b)^2.$$

We choose

$$b = \min\left(\frac{1}{2\beta}, \frac{1}{6}\right) = \frac{1}{2\max(\beta, 3)}. \tag{G9}$$

In particular, $0 \leqslant b \leqslant \frac{1}{6}$ ensures that $(1-2b)^2 \geqslant 2b(1-b)$.

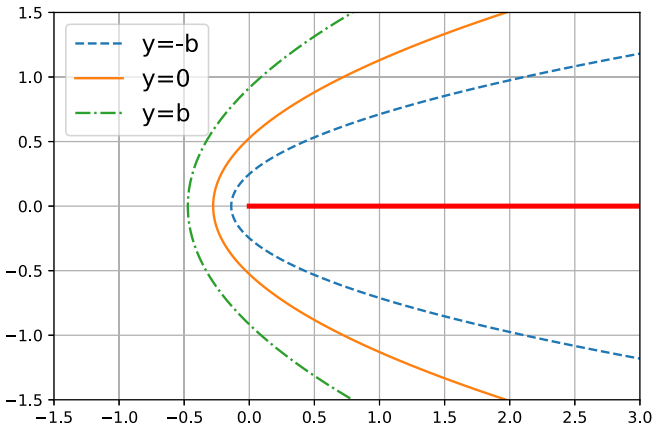


FIG. 6. The parabola $\{t^2 + it - \zeta : \operatorname{Im} t = y\}$ for $y = -b, 0, b$. The spectrum of $H = A + B$ is on the positive part of the real axis (solid red line).

We consider two cases. First, when $w^2 \geqslant b(1-b)$, we have

$$\begin{aligned}[w^2 - b(1-b) - x]^2 + w^2(1-2b)^2 &\geqslant w^2(1-2b)^2 \\ &\geqslant b(1-b)(1-2b)^2 \geqslant 2b^2(1-b)^2.\end{aligned}$$

Second, when $w^2 < b(1-b)$, because $x \geqslant 0$, we have

$$\begin{aligned}&[w^2 - b(1-b) - x]^2 + w^2(1-2b)^2 \\ &\quad = w^4 + [(1-2b)^2 - 2b(1-b)]w^2 + b^2(1-b)^2 \\ &\quad \geqslant b^2(1-b)^2.\end{aligned}$$

Combining these two cases we have

$$|t^2 + it - x - \zeta| \geqslant b(1-b) = \frac{\zeta}{2} \tag{G10}$$

for all $|\operatorname{Im} t| \leqslant b$ and $x \geqslant 0$.

Next, we determine how far $h(\theta, x)$ can be analytically extended. By the relation $t = T\cos(\theta)$, we have

$$\operatorname{Im} t = -T\sin(\operatorname{Re}\theta)\sinh(\operatorname{Im}\theta).$$

Therefore,

$$|\operatorname{Im} t| \leqslant T\sinh(|\operatorname{Im}\theta|) \leqslant 2T|\operatorname{Im}\theta|$$

when $|\operatorname{Im}\theta| \leqslant 1$. Note that so far we have only used $b \leqslant \frac{1}{6}$. In Eq. (G9) we also have $b \leqslant 1/(2\beta)$, which ensures that $|e^{-\beta z}|$ for all z in a vicinity of the parabola can be bounded from above by a constant. This is very important when we prove the exponential decay of the Fourier coefficients below.

Therefore, for $|\operatorname{Im} t| \leqslant b$ we only need to require $|\operatorname{Im}\theta| \leqslant b/2T$ when $T \geqslant \frac{1}{2}$. Thus, $h(\theta, x)$ can be analytically extended in θ to the strip $\{\theta \in \mathbb{C} : |\operatorname{Im}\theta| \leqslant b/2T\}$. Also, $h(\theta, x)$ is periodic in the real direction with a period 2π. Now consider $\widetilde{h}(z, x) = h(\theta, x)$ with $z = e^{i\theta}$, then $\widetilde{h}(\cdot, x)$ is analytic in the annulus $\{z \in \mathbb{C} : e^{-b/2T} < |z| < e^{b/2T}\}$. The coefficients of the Laurent series can be evaluated using the residue theorem as

$$\hat{h}_n(x) = \frac{1}{2\pi i} \oint_{\Gamma(a)} z^{-(n+1)} \widetilde{h}(z, x) dz,$$

where the contour is $\Gamma(a) := \{z : |z| = e^a\}$ with $-b/2T < a < b/2T$. By changing the variable back from z to θ, we

arrive at the following bounds for the Fourier coefficients:

$$|\hat{h}_n(x)| \leqslant \frac{1}{2\pi e^{a|n|}} \int_0^{2\pi} |h(\theta - ia, x)| d\theta, \quad n \geqslant 0,$$
$$\|\hat{h}_n(x)| \leqslant \frac{1}{2\pi e^{a|n|}} \int_0^{2\pi} |h(\theta + ia, x)| d\theta, \quad n < 0, \quad \text{(G11)}$$

for $0 < a < b/2T$.

We then bound $h(\theta + ia, x)$ for $|a| < b/2T$. Let $t = w + iy$, then as analyzed above $|a| < b/2T$ guarantees $|y| \leqslant b$. Hence, $\mathrm{Re}(t^2 + it)$ is minimized when $w = 0$ and $y = b$. In this case $\mathrm{Re}(t^2 + it) = -b(1 + b)$. Thus, we have

$$\mathrm{Re}(t^2 - \zeta + it) \geqslant -b(1 + b) - \zeta = b(1 - 3b) \geqslant \frac{1}{4\beta}, \quad \text{(G12)}$$

using the fact $b \leqslant \frac{1}{6}$ and $b \leqslant 1/(2\beta)$ in Eq. (G9). This enables us to bound the exponential in $g(t, x)$. We combine the bound for the exponential with Eq. (G10) to ensure

$$|h(\theta + ia, x)| = |g(t, x)| = \left| \frac{e^{-\beta(t^2 - \zeta + it)}(2t + i)}{x - t^2 + \zeta - it} \right| \leqslant \frac{8Te^{1/4}}{\zeta}.$$

We have chosen b and ζ in Eqs. (G9) and (G8), respectively. Using these two equations and the fact that $b \leqslant \frac{1}{6} < \frac{1}{2}$ we have $1/\zeta \leqslant 1/b = 2\max(\beta, 3)$. Therefore,

$$|h(\theta + ia, x)| \leqslant 16Te^{1/4} \max(\beta, 3).$$

Taking it into Eq. (G11) we have

$$|\hat{h}_n(x)| \leqslant \frac{16T \max(\beta, 3) e^{1/4}}{e^{a|n|}}$$

for $0 < a < b/2T = 1/[4T \max(\beta, 3)]$ and $n \in \mathbb{Z}$. Using this to get $\hat{g}_n(x)$, setting $a = 1/[8T \max(\beta, 3)]$, and taking into Eq. (G7), we have

$$|I_T - I_{\mathrm{GL}}| \leqslant \sum_{n \geqslant 2J} \frac{64T^2 \widetilde{\beta} e^{1/4}}{e^{an}} = \frac{64T^2 \widetilde{\beta} e^{1/4}}{1 - e^{-1/(8T\widetilde{\beta})}} e^{-J/(4T\widetilde{\beta})}, \quad \text{(G13)}$$

where $\widetilde{\beta} = \max(\beta, 3)$. Combining this inequality with (G4), in which we use $\beta\zeta \leqslant 1$, we have

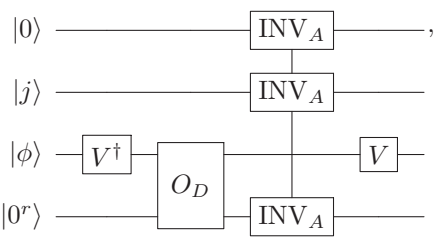
$$|I - I_{\mathrm{GL}}| \leqslant |I - I_T| + |I_T - I_{\mathrm{GL}}| \leqslant \sqrt{\frac{2}{\beta\pi}} e^{1 - \beta T^2} + \frac{64T^2 \widetilde{\beta} e^{1/4}}{1 - e^{-1/(8T\widetilde{\beta})}} e^{-J/(4T\widetilde{\beta})}. \quad \text{(G14)}$$

APPENDIX H: EFFICIENT BLOCK ENCODINGS FOR THE CONTOUR-INTEGRAL APPROACH

In this Appendix we construct the block encodings in Lemma 3. The first thing we need to implement is the block-diagonal matrix

$$\sum_{j \in [J]} |j\rangle\langle j| \otimes (z_j + \xi_j - A)^{-1} = \begin{pmatrix} (z_1 + \xi_1 - A)^{-1} & & & \\ & (z_2 + \xi_2 - A)^{-1} & & \\ & & \ddots & \\ & & & (z_J + \xi_J - A)^{-1} \end{pmatrix}. \quad \text{(H1)}$$

This block-diagonal matrix can be implemented by the following circuit:



(H2)

where INV_A satisfies

$$\mathrm{INV}_A|0\rangle|j\rangle|\lambda\rangle = \left(\frac{1}{z_j + \xi_j - \lambda}|0\rangle + \sqrt{1 - \left| \frac{1}{z_j + \xi_j - \lambda} \right|^2} |1\rangle \right) |j\rangle|\lambda\rangle.$$

This is a $(1, O(r + \ln(J)), 0)$ block encoding of the block-diagonal matrix in Eq. (H1). The main purpose of the above design is to use V and O_D only $O(1)$ times instead of $O(J)$ times.

Similar to the above construction, it is possible to construct a $(1 + \alpha_B, O(1), 0)$ block encoding of the following block-diagonal matrix:

$$\sum_{j \in [J]} |j\rangle\langle j| \otimes (B + \xi_j) = \begin{pmatrix} B + \xi_1 & & & \\ & B + \xi_2 & & \\ & & \ddots & \\ & & & B + \xi_J \end{pmatrix}, \quad \text{(H3)}$$

using controlled U_B only once and a logical circuit to determine the value of ξ_j for each input j.

APPENDIX I: PROOF OF THEOREM 4: CONVERGENCE OF THE TRUNCATED CHEBYSHEV SERIES

In this Appendix we prove Theorem 4. For completeness we repeat the essential steps of the proof in [66, Sec. 5.7] with a more explicit constant dependence, which is the key to obtain exponential convergence of functions in the Gevrey class.

Using Peano's theorem [66, Lemma 5.15],

$$\sum_{k=0}^{d} c_k T_k(y) - g(y) = \int_{-1}^{1} g^{(r+1)}(t) K_d(y,t) dt. \tag{I1}$$

Here

$$K_d(y,t) = \frac{1}{r!}\left(\sum_{j=0}^{d} c_{jr} T_j(y) - (y-t)_+^r\right), \tag{I2}$$

where

$$(y-t)_+^r := \begin{cases} (y-t)^r, & y \geqslant t \\ 0, & y < t \end{cases}$$

and

$$c_{jr} = \frac{2-\delta_{j0}}{\pi} \int_t^1 \frac{(y-t)^r T_j(y)}{\sqrt{1-y^2}} dy.$$

Note that K_d does not depend on the function $g(y)$.

Now let us bound the coefficient c_{jr} for all $j \geqslant 1$. Using the substitution $y = \cos\theta$ and $t = \cos\phi$,

$$c_{jr} = \frac{2}{\pi} \int_0^{\phi} [\cos(\theta) - \cos(\phi)]^r \cos(j\theta) d\theta. \tag{I3}$$

Define $h(s) = [s - \cos(\phi)]^r$ and $f(\theta) = h(\cos(\theta))$. Notice that the $(l+1)$th-order antiderivative of $\cos(j\theta)$ is $\frac{(-1)^{l+1}}{j^{l+1}} \cos^{(l+1)}(j\theta)$, using integration by parts, we obtain

$$\frac{\pi}{2} c_{jr} = -\left[\sum_{l=0}^{r} f^{(l)}(\theta) \frac{1}{j^{l+1}} \cos^{(l+1)}(j\theta)\right]_0^{\phi} + \frac{1}{j^{r+1}} \int_0^{\phi} f^{(r+1)}(\theta) \cos^{(r+1)}(j\theta) d\theta. \tag{I4}$$

By Lemma 5,

$$f^{(l)}(\theta) = \sum_{\sum_{p=1}^{l} p q_p = l} \frac{l!}{q_1!(1!)^{q_1} q_2!(2!)^{q_2} \dots q_l!(l!)^{q_l}} h^{(q_1+q_2+\dots+q_l)}[\cos(\theta)] \prod_{p=1}^{l} [\cos^{(p)}(\theta)]^{q_p} \tag{I5}$$

$$= \sum_{\sum_{p=1}^{l} p q_p = l} \frac{l!}{q_1!(1!)^{q_1} q_2!(2!)^{q_2} \dots q_l!(l!)^{q_l}} \frac{r!}{(r - \sum_p q_p)!} [\cos(\theta) - \cos(\phi)]^{r - \sum_p q_p} \prod_{p=1}^{l} [\cos^{(p)}(\theta)]^{q_p}. \tag{I6}$$

Notice that for all $l \leqslant r-1$, $\sum_p q_p \leqslant \sum_p p q_p = l < r$, we have $f^{(l)}(\phi) = 0$ for all $l \leqslant r-1$. Then,

$$\frac{\pi}{2} c_{jr} = -f^{(r)}(\phi) \frac{1}{j^{r+1}} \cos^{(r+1)}(j\phi) + \sum_{l=0}^{r} f^{(l)}(0) \frac{1}{j^{l+1}} \cos^{(l+1)}(0) + \frac{1}{j^{r+1}} \int_0^{\phi} f^{(r+1)}(\theta) \cos^{(r+1)}(j\theta) d\theta.$$

Furthermore, when l is odd, from the equation $\sum_{j=1}^{l} j q_j = l$, for any tuple $(q_1, \dots, q_l)$, there must exist an odd number p_0, such that $q_{p_0} \neq 0$. Therefore, $[\cos^{(p_0)}(0)]^{q_{p_0}} = 0$, and thus $f^{(l)}(0) = 0$. When l is even, $\cos^{(l+1)}(0) = 0$. Therefore, we have, for all $l \leqslant r$, $f^{(l)}(0) \cos^{(l+1)}(0) = 0$, and

$$\frac{\pi}{2} c_{jr} = -f^{(r)}(\phi) \frac{1}{j^{r+1}} \cos^{(r+1)}(j\phi) + \frac{1}{j^{r+1}} \int_0^{\phi} f^{(r+1)}(\theta) \cos^{(r+1)}(j\theta) d\theta. \tag{I7}$$

Finally, using some very rough estimates that $|\cos^{(p)}(\theta)| \leqslant 1$ and dropping all the denominators in $f^{(l)}(\theta)$ (which can be surely improved, but here for technical simplicity we keep these rough estimates), and that the number of l-tuples is less than $(l+1)(l/2+1)(l/3+1)\dots(l/l+1) = \binom{2l}{l} < 2^{2l}$, we have

$$\|f^{(l)}\|_{\infty} \leqslant 2^{2l} l! r! 2^r = 2^{2l+r} l! r!, \tag{I8}$$

and we obtain

$$|c_{jr}| \leqslant \frac{2}{\pi}\frac{2^{3r}(r!)^2}{j^{r+1}} + \frac{2}{\pi}\frac{\pi 2^{3r+2}(r+1)!r!}{j^{r+1}} \leqslant 2^{3r+4}\frac{(r+1)!r!}{j^{r+1}}.$$

It follows that

$$\begin{aligned}|K_d(y,t)| &\leqslant \frac{1}{r!}\left|\sum_{j=d+1}^{\infty} c_{jr}T_j(y)\right| \leqslant \frac{1}{r!}\sum_{j=d+1}^{\infty}|c_{jr}| \\ &\leqslant \frac{2^{3r+4}(r+1)!r!}{r!}\sum_{j=d+1}^{\infty}\frac{1}{j^{r+1}} \\ &< 2^{3r+4}(r+1)!\int_d^{\infty}\frac{1}{x^{r+1}}dx \leqslant 16\frac{8^r(r+1)!}{d^r}.\end{aligned}$$

Combining this estimate with Eq. (I1), we complete the proof.

APPENDIX J: GREEN'S-FUNCTION COMPUTATION FOR FIXED NUMBER OF ELECTRONS

Here we review how the scaling of the complexity for evaluating Green's functions for certain quantum many-body Hamiltonians can be improved by utilizing the electron-number constraint as in Sec. IV C. Our aim in this section is to discuss how the block-encoding construction can be modified in these cases to accommodate this.

First, let us assume that we have a Hamiltonian that is the sum of two orthogonally diagonalizable Hamiltonians. This means that if we express the Hamiltonian in its eigenbasis then there exists a unitary matrix $\hat{U}$ such that $H = H_0 + A := \hat{U}D\hat{U}^\dagger + A$ such that A and D are diagonal matrices and $\hat{U}$ transforms from the computational basis to the eigenbasis of H_0. Specifically, if we let $|\psi_k\rangle$ be an eigenstate of H_0 then

$$\hat{U}|\psi_k\rangle = |k\rangle, \tag{J1}$$

where $|k\rangle$ is the kth computational basis vector.

If we let P_{N_e} be a projector onto a constrained manifold, in our case the manifold of states with a fixed electron number N_e, then we can define the constrained Hamiltonian H' via

$$H' = P_{N_e}(\hat{U}D\hat{U}^\dagger + A)P_{N_e}. \tag{J2}$$

Further, let us also assume that P_{N_e} commutes not only with H but also A. If this is true, then $[H', P_{N_e}] = [\hat{U}D\hat{U}^\dagger, P_{N_e}] = 0$. Therefore, $\hat{U}D\hat{U}^\dagger P_{N_e} = P_{N_e}\hat{U}D\hat{U}^\dagger$ and in turn

$$H' = (\hat{U}(\hat{U}^\dagger P_{N_e}\hat{U})D\hat{U}^\dagger) + AP_{N_e} \tag{J3}$$

has the same action within the subspace conditioned on the value of N_e. Specifically, for any $+1$ eigenstate of P_{N_e} denoted by $|\psi\rangle$, we have

$$H|\psi\rangle = HP_{N_e}|\psi\rangle = H'|\psi\rangle. \tag{J4}$$

This validates the claim that such modifications do not affect the action of the Hamiltonian H' on the fixed-particle manifold.

Furthermore, let $H'' = H' + C(1 - P_{N_e})$ for Hermitian matrix C. We then have that for any $+1$ eigenstate of P_{N_e} $|\psi\rangle$

$$H''|\psi\rangle = H''P_{N_e}|\psi\rangle = H'|\psi\rangle. \tag{J5}$$

Thus, we can perturb H any way we see fit so long as the perturbation has no impact on the particle-number manifold in question. We will use this fact to simplify our block-encoding construction.

Since $H_0 = UDU^\dagger$, we simply need to construct a unitary for block encoding D in order to e convert this classical circuit into a quantum circuit O_D satisfying

$$O_D|k\rangle|c\rangle = |k\rangle|c \oplus D_{kk}\rangle.$$

For each eigenstate $|\psi_k\rangle$ of H_0 indexed by k, we can compute the particle number efficiently through a classical circuit, which we also convert to a quantum circuit O_{num}, that satisfies

$$O_{\text{num}}|k\rangle|c\rangle = |k\rangle|c \oplus N_e(k)\rangle,$$

where $\hat{N}$ is the number operator and

$$N_e(k) = \langle\psi_k|\hat{N}|\psi_k\rangle.$$

The components work as follows:

$$\text{CMP}_{\text{num}}|N_e(k)\rangle|c\rangle = \begin{cases}|N_e(k)\rangle|c \oplus 1\rangle, & \text{if } N_e(k) = N_e \\ |N_e(k)\rangle|c\rangle, & \text{if } N_e(k) \neq N_e\end{cases}$$

and

$$\begin{aligned}&R|D_{kk}\rangle|c\rangle|0\rangle \\ &= \begin{cases}|D_{kk}\rangle|c\rangle\left(\frac{D_{kk}}{\alpha(N_e)}|0\rangle + \sqrt{1-\left(\frac{D_{kk}}{\alpha(N_e)}\right)^2}|1\rangle\right), & \text{if } c = 0 \\ |D_{kk}\rangle|c\rangle|1\rangle, & \text{if } c = 1.\end{cases}\end{aligned}$$

We only need to make sure $\alpha(N_e) \geqslant D_{kk}$ for all k such that $N_e(k) = N_e$. Therefore, in the case of the kinetic operator that appears in the Hubbard model, this gives a block encoding of $H_0P_{N_e}$ with subnormalization factor $\alpha(N_e)$.

Next we will assume that $[\hat{U}, P_{N_e}] = 0$. While this assumption is not strictly needed to simplify the Hamiltonian to accommodate the particle-number constraint, the algorithmic design will be easier. As an example, consider the fermionic Fourier transform

$$\begin{aligned}\left[\text{FFFT}, \sum_j \hat{n}_j\right] &= \text{FFFT}\sum_j \hat{n}_j - \sum_j \hat{n}_j\text{FFFT} \\ &= \text{FFFT}\left(\sum_j \hat{n}_j - \sum_j \hat{c}_j^\dagger\hat{c}_j\right) \\ &= \sum_k \text{FFFT}\left(\sum_k kP_k - \sum_k kP_k\right) = 0.\end{aligned} \tag{J6}$$

Therefore, we have that $[\text{FFFT}, P_{N_e}] = 0$ and in turn this holds for the Hubbard model, plane-wave dual simulations, and the Schwinger model. In all these cases it follows that

$$H' = \hat{U}(P_{N_e}D)\hat{U}^\dagger + AP_{N_e}. \tag{J7}$$

Thus, in these cases we can zero out any eigenvalues of the orthogonally diagonalizable operators that are outside of the constraint specified by P_{N_e}.

For the case of Green's-function calculation for the Hubbard model, $\hat{U} = \text{FFFT}$ and we can implement a block encoding for $H_0P_{N_e}$ on the fixed particle-number block using the construction in Fig. 7. In particular, from (22) it is clear that if we project the state onto the manifold consisting of N_e electrons we obtain using the fact that FFFT commutes with

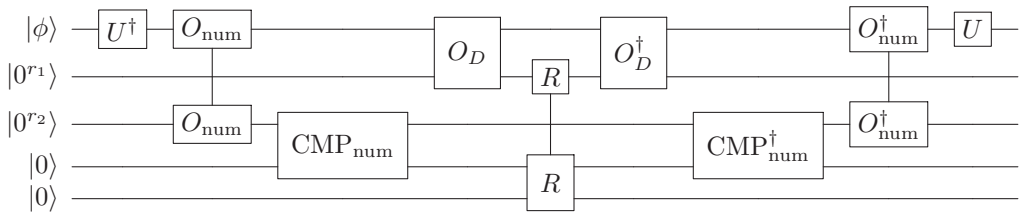


FIG. 7. Circuit for performing a block-encoding projector on a fixed particle-number manifold of states.

the projector onto the manifold with N_e electrons (denoted P_{N_e}) we observe that

$$\sum_{\mathbf{x},\mathbf{y},\sigma} T(\mathbf{x}-\mathbf{y})\hat{a}^\dagger_{\mathbf{x}\sigma}\hat{a}_{\mathbf{y}\sigma}P_{N_e} = \text{FFFT}\sum_{\mathbf{G},\sigma}\hat{T}(\mathbf{G})\hat{c}^\dagger_{\mathbf{G},\sigma}\hat{c}_{\mathbf{G},\sigma}P_{N_e}\text{FFFT}^\dagger. \tag{J8}$$

We therefore have that the normalization factor for the block encoding of the kinetic operator constrained to this subspace obeys

$$\alpha_T \leqslant \frac{N_e}{2}\max_{\mathbf{G}}|\hat{T}(\mathbf{G})| = \max_{k_x,k_y}\left|\frac{N_e}{2N}\sum_{x,y}T(x,y)e^{2\pi i(k_x x+k_y y)/\sqrt{N}}\right| \leqslant N_e|t|. \tag{J9}$$

The argument for α_U is exactly the same except we do not need to worry about the diagonalizing FFFT operation. This means that we can directly apply the reasoning in (24) to find

$$\alpha_U \leqslant N_e|U|. \tag{J10}$$

Thus, by choosing our preconditioner appropriately, we can achieve $\alpha_B \in O(N_e \min(|t|, |U|))$. The reasoning for computing Green's functions for the Coulomb interaction in the plane-wave dual basis and the Schwinger model follows identically.

As a final note, this block-encoding technique is not just specific to Green's-function evaluation. Hamiltonian simulation and other related tasks can also be improved in the continuum limit by using this strategy.

[1] A. Ambainis, Variable time amplitude amplification and quantum algorithms for linear algebra problems, in *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)* (Springer, Berlin, 2012), Vol. 14, pp. 636–647.

[2] D. An and L. Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, arXiv:1909.05500.

[3] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, Variational quantum linear solver: A hybrid algorithm for linear systems, arXiv:1909.05820.

[4] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the Poisson equation, New J. Phys. **15**, 013021 (2013).

[5] S. Chakraborty, A. Gilyén, and S. Jeffery, The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs) Vol. 132 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), pp. 33:1–33:14.

[6] 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).

[7] 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 Theory of Computing* (ACM, New York, 2019), pp. 193–204.

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

[9] L. Lin and Y. Tong, Optimal quantum eigenstate filtering with application to solving quantum linear systems, Quantum **4**, 361 (2020).

[10] Y. Subaşı, R. D. Somma, and D. Orsucci, Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing, Phys. Rev. Lett. **122**, 060504 (2019).

[11] L. Wossnig, Z. Zhao, and A. Prakash, Quantum Linear System Algorithm for Dense Matrices, Phys. Rev. Lett. **120**, 050502 (2018).

[12] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational algorithms for linear algebra, arXiv:1909.03898.

[13] Y. Saad, *Iterative Methods for Sparse Linear Systems* (SIAM, Philadelphia, 2003), Vol. 82.

[14] C.-J. Lin and J. J. Moré, Incomplete Cholesky factorizations with limited memory, SIAM J. Sci. Comput. **21**, 24 (1999).

[15] T. A. Manteuffel, An incomplete factorization technique for positive definite linear systems, Math. Comput. **34**, 473 (1980).

[16] J. A. Meijerink and H. A. Van Der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric *m*-matrix, Math. Comput. **31**, 148 (1977).

[17] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Preconditioned Quantum Linear System Algorithm, Phys. Rev. Lett. **110**, 250504 (2013).

[18] N. Higham, *Functions of Matrices: Theory and Computation*, Vol. 104 (SIAM, Philadelphia, 2008).

[19] D. Poulin and P. Wocjan, Preparing Ground States of Quantum Many-Body Systems on a Quantum Computer, Phys. Rev. Lett. **102**, 130503 (2009).

[20] M. Kieferová and N. Wiebe, Tomography and generative training with quantum Boltzmann machines, Phys. Rev. A **96**, 062327 (2017).

[21] F. G. Brandao and K. M. Svore, Quantum speed-ups for solving semidefinite programs, in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Piscataway, NJ, 2017), pp. 415–426.

[22] E. Tang, Quantum-Inspired Classical Algorithms for Principal Component Analysis and Supervised Clustering, Phys. Rev. Lett. **127**, 060503 (2018).

[23] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (ACM, New York, 2019), pp. 217–228.

[24] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).

[25] C. Shao and H. Xiang, Quantum circulant preconditioner for a linear system of equations, Phys. Rev. A **98**, 062321 (2018).

[26] M. Benzi and M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, SIAM J. Sci. Comput. **19**, 968 (1998).

[27] M. Benzi, C. Meyer, and M. Tuma, A sparse approximate inverse preconditioner for the conjugate gradient method, SIAM J. Sci. Comput. **17**, 1135 (1996).

[28] G. R. Ahokas, Improved algorithms for approximate quantum Fourier transforms and sparse Hamiltonian simulations, Master's thesis, University of Calgary, 2004.

[29] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Exponential algorithmic speedup by a quantum walk, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing* (ACM, New York, 2003), pp. 59–68.

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

[31] R. M. Martin, L. Reining, and D. M. Ceperley, *Interacting Electrons* (Cambridge University Press, Cambridge, 2016).

[32] J. W. Negele and H. Orland, *Quantum Many-particle Systems* (Westview Press, Boulder, CO, 1988).

[33] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Hybrid Quantum-Classical Approach to Correlated Materials, Phys. Rev. X **6**, 031045 (2016).

[34] S. Endo, I. Kurata, and Y. O. Nakagawa, Calculation of the Green's function on near-term quantum computers, Phys. Rev. Research **2**, 033281 (2020).

[35] X. Cai, W.-H. Fang, H. Fan, and Z. Li, Quantum computation of molecular response properties, Phys. Rev. Research **2**, 033324 (2020).

[36] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 2015), pp. 792–809.

[37] G. H. Low and I. L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, Phys. Rev. Lett. **118**, 010501 (2017).

[38] D. Poulin and P. Wocjan, Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer, Phys. Rev. Lett. **103**, 220502 (2009).

[39] J. Van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, Quantum SDP-solvers: Better upper and lower bounds, Quantum **4**, 230 (2020).

[40] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[41] M. Ozols, M. Roetteler, and J. Roland, Quantum rejection sampling, ACM Trans. Comput. Theory (TOCT) **5**, 1 (2013).

[42] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Quantum metropolis sampling, Nature (London) **471**, 87 (2011).

[43] M.-H. Yung and A. Aspuru-Guzik, A quantum–quantum Metropolis algorithm, Proc. Natl. Acad. Sci. USA **109**, 754 (2012).

[44] A. N. Chowdhury and R. D. Somma, Quantum algorithms for Gibbs sampling and hitting-time estimation, Quantum Inf. Comput. **17**, 0041 (2017).

[45] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, arXiv:1806.01838.

[46] S. Arora and B. Barak, *Computational Complexity: A Modern Approach* (Cambridge University Press, Cambridge, 2009).

[47] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[48] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[49] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe, The power of quantum systems on a line, Commun. Math. Phys. **287**, 41 (2009).

[50] J. Kempe, A. Kitaev, and O. Regev, The complexity of the local Hamiltonian problem, SIAM J. Comput. **35**, 1070 (2006).

[51] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and Quantum Computation*, Number 47 (American Mathematical Society, Providence, RI, 2002).

[52] R. Oliveira and B. M. Terhal, The complexity of quantum spin systems on a two-dimensional square lattice, Quantum Inf. Comput. **8**, 0900 (2008).

[53] Y. Ge, J. Tura, and J. I. Cirac, Faster ground state preparation and high-precision ground energy estimation with fewer qubits, J. Math. Phys. **60**, 022202 (2019).

[54] L. Lin and Y. Tong, Near-optimal ground state preparation, Quantum **4**, 372 (2020).

[55] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-Depth Quantum Simulation of Materials, Phys. Rev. X **8**, 011044 (2018).

[56] E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Continuous-time Monte Carlo methods for quantum impurity models, Rev. Mod. Phys. **83**, 349 (2011).

[57] L. M. Fraser, W. M. C. Foulkes, G. Rajagopal, R. Needs, S. Kenny, and A. Williamson, Finite-size effects and Coulomb interactions in quantum Monte Carlo calculations for homogeneous systems with periodic boundary conditions, Phys. Rev. B **53**, 1814 (1996).

[58] J. Kogut and L. Susskind, Hamiltonian formulation of Wilson's lattice gauge theories, Phys. Rev. D **11**, 395 (1975).

[59] A. F. Shaw, P. Lougovski, J. R. Stryker, and N. Wiebe, Quantum algorithms for simulating the lattice Schwinger model, Quantum **4**, 306 (2020).

[60] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, Talbot quadratures and rational approximations, BIT Numer. Math. **46**, 653 (2006).

[61] W. Cody, G. Meinardus, and R. Varga, Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems, J. Approximation Theory **2**, 50 (1969).

[62] R. Piessens, Gaussian quadrature formulas for the numerical integration of Bromwich's integral and the inversion of the Laplace transform, J. Eng. Math. **5**, 1 (1971).

[63] A. Talbot, The accurate numerical inversion of Laplace transforms, IMA J. Appl. Math. **23**, 97 (1979).

[64] J. Weideman and L. Trefethen, Parabolic and hyperbolic contours for computing the Bromwich integral, Math. Comput. **76**, 1341 (2007).

[65] S. G. Krantz and H. R. Parks, *A Primer of Real Analytic Functions* (Springer, New York, 2002).

[66] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials* (Chapman & Hall, New York, 2003).

[67] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum deep learning, Quantum Inf. Comput. **16**, 541 (2016).

[68] G. H. Low and N. Wiebe, Hamiltonian simulation in the interaction picture, arXiv:1805.00675.

[69] G. H. Low, T. J. Yoder, and I. L. Chuang, Methodology of Resonant Equiangular Composite Quantum Gates, Phys. Rev. X **6**, 041067 (2016).

[70] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[71] S. Jansen, M.-B. Ruskai, and R. Seiler, Bounds for the adiabatic approximation with applications to quantum computation, J. Math. Phys. **48**, 102111 (2007).

[72] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv:quant-ph/0001106.

[73] S. Boixo, E. Knill, and R. D. Somma, Eigenpath traversal by phase randomization, Quantum Inf. Comput. **9**, 833 (2009).

[74] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028

[75] F. Arrigo, M. Benzi, and C. Fenu, Computation of generalized matrix functions, SIAM J. Matrix Anal. Appl. **37**, 836 (2016).

[76] J. B. Hawkins and A. Ben-Israel, On generalized matrix functions, Linear and Multilinear Algebra **1**, 163 (1973).

[77] R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy, Finding angles for quantum signal processing with machine precision, arXiv:2003.02831.

[78] Y. Dong, X. Meng, K. B. Whaley, and L. Lin, Efficient phase factor evaluation in quantum signal processing, Phys. Rev. A **103**, 042419 (2021).

[79] J. Haah, Product decomposition of periodic functions in quantum signal processing, Quantum **3**, 190 (2019).

[80] Y. R. Sanders, D. W. Berry, P. Costa, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, Compilation of fault-tolerant quantum heuristics for combinatorial optimization, PRX Quantum **1**, 020312 (2020).

[81] 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* (ACM, New York, 2003), pp. 20–29.

[82] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, Phys. Rev. Lett. **114**, 090502 (2015).

[83] M. Chiani, D. Dardari, and M. K. Simon, New exponential bounds and approximations for the computation of error probability in fading channels, IEEE Trans. Wireless Commun. **2**, 840 (2003).

[84] L. N. Trefethen, *Approximation Theory and Approximation Practice* (SIAM, Philadelphia, 2019), Vol. 164.