Quantum Eigenstate Filtering and Its Applications

by

Yu Tong

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Lin Lin, Chair
Professor Eric Neuscamman
Professor Panayiotis Papadopoulos
Professor Jon Wilkening

Summer 2022

Quantum Eigenstate Filtering and Its Applications

Abstract

Quantum Eigenstate Filtering and Its Applications

by

Yu Tong

Doctor of Philosophy in Applied Mathematics

University of California, Berkeley

Professor Lin Lin, Chair

Recent years have seen rapid progress in the field of quantum algorithms for linear algebra problems. These algorithms can solve important problems in quantum chemistry, condensed matter physics, and quantum field theory simulation, with potentially exponential speedup compared to classical algorithms. The progress is in part due to the development of new methods to implement matrix functions on a quantum computer. Examples include the linear combination of unitaries (LCU) method, quantum signal processing (QSP), and quantum singular value transformation (QSVT), which is closely related to QSP. Using these methods, one can construct matrix functions to filter out unwanted eigenstates of a given Hermitian matrix, and thereby obtain a target eigenstate on a quantum computer. This technique we call *quantum eigenstate filtering.*

We focus our attention on two specific problems: ground state preparation and energy estimation, and solving linear systems. For the ground state preparation problem, we want to prepare the ground state, i.e., the lowest eigenstate of the Hamiltonian H, while for ground state energy estimation, we want to estimate the ground state energy, i.e., the the lowest eigenvalue of the Hamiltonian H. The ground state energy estimation problem is cannot be solved in polynomial time even on a quantum computer in the worst case. However, this problem becomes tractable with additional assumptions, which may be satisfied in many real-world applications. In this work we assume access to a good initial guess of the ground state, such that its overlap with the ground state is lower bounded by a parameter γ. Under this assumption, we present quantum algorithms based on the quantum eigenstate filtering technique to estimate the ground state energy and to prepare the ground state, with nearly

optimal scaling with respect to both γ and the precision.

The above mentioned algorithms work in the setting where the quantum computers are fully fault-tolerant, i.e., the noise is sufficiently suppressed through quantum error correction. We also present an algorithm to estimate the ground state energy in the *early fault-tolerant setting*, in which the the algorithm is limited by the number of qubits available and the coherence time. In this setting our algorithm uses a simple circuit and lower circuit depth at the expense of longer runtime. However the algorithm can still achieve the optimal Heisenberg-limited precision scaling despite these restrictions.

The quantum eigenstate filtering technique is also useful in solving the quantum linear system problem, in which we are asked to solve $Ax = b$ on a quantum computer. We show that the problem can be solved with a query complexity almost matching its lower bound by traversing the adiabatic path using quantum eigenstate filtering.

Chapter 1 introduces the basic problem setup and provides an overview of the results in this paper. Chapter 2 presents the nearly optimal algorithms for ground state preparation and energy estimation. Chapter 3 discussed implementation on early fault-tolerant devices. Chapter 4 applies quantum eigenstate filtering to solve the quantum linear system problem. Chapters 2, 3, 4 are based on [Lin, Tong, Quantum 4 (2020), p. 372], [Lin, Tong, PRX Quantum 3.1 (2022), p. 010318], [Lin, Tong, Quantum 4 (2020), p. 361] respectively, all of which are published under CC BY 4.0.

Contents

Acknowledgments

First I would like to thank my advisor Lin Lin for his support and guidance throughout my Ph.D. years. His enthusiasm for constantly exploring new problems and learning new things has always been a great inspiration to me, and his generous help has been indispensable at every stage of my Ph.D. journey.

In my Ph.D. years I have had the wonderful opportunity to collaborate with a number of people who helped me grow as a researcher. I would like to thank John Preskill for inspiring discussions when I was struggling to broaden the scope of my research, Nathan Wiebe for introducing me to many topics in quantum computing, Jarrod McClean for hosting me during my internship at Google Quantum AI, and Yuan Su for fruitful collaboration and good career advice.

At UC Berkeley I met a group of people who, through discussion and collaboration, gave me much inspiration and intellectual enjoyment. Among them I would like to thank especially Michael Lindsey, who helped me to get to know the quantum many-body problem, which has become a kind of grand goal of all my research projects, and Dong An, who helped me immensely in learning and developing quantum algorithms. I would also like to thank other members in Lin's group for sharing their expertise and insights in discussions, and for having a good time together.

It is impossible to properly thank all the people whose friendship and good will have helped me along the way. Here, let me express my gratitude to everyone who has supported and encouraged me. Finally I would like to thank my girlfriend Di for being there for me whenever I need it.

Chapter 1

Introduction

Recent advances in quantum computing technologies have given rise to prospects for using quantum computers and quantum algorithms to achieve significant speedup in certain important tasks. A major motivation of the development of quantum computing technologies is the fact that a fault-tolerant quantum computer can factor large integers exponentially faster than any known classical algorithms [147]. However, there are other tasks, which may more directly further understanding of nature, that quantum computers can potentially perform much more efficiently than classical computers. Among these tasks are the simulation of systems that are by nature quantum mechanical, which is the focus of this work.

The quantum systems we consider come from quantum chemistry [23, 120, 28, 104, 153], condensed matter physics [4, 103], and also from spatially discretizing a quantum field theory [99, 137, 162]. The simulation tasks can be roughly divided into two categories: static simulation and dynamic simulation. For the former, our goal is to estimate observable expectation values in the ground state or the thermal state. In this setting, we are particularly interested in the ground state energy. For the latter, we want to estimate expectation values during real-time evolution. We will mainly focus on the first task in this work, but will also briefly discuss the second task.

Many classical algorithms have been developed for the first task, but none of the current classical method can obtain the ground state energy with polynomial precision, with a runtime that is polynomial in the system size, and there is good reason from a complexity theory point of view that no such classical or quantum algorithm can exist for the generic case, because this problem is shown to be QMA-hard [102, 100, 130, 6]. However, under the assumption that we have an initial state with an inverse polynomially large overlap[1] with the ground state, it is possible for a quantum computer to attain polynomial precision in polynomial time.

Let us state the problem more formally. We consider a quantum system consisting of n qubits, whose state can be described by a vector in the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$. The Hilbert space dimension is therefore $N = 2^n$. The evolution of the quantum system is governed by a Hamiltonian H, which is a Hermitian operator on the Hilbert space $\mathcal{H}$. We can write down the eigenpairs of H as $(\lambda_k, |\psi_k\rangle)$, with

$$H\,|\psi_k\rangle = \lambda_k\,|\psi_k\rangle\,.$$

The eigenvalues are arranged in the ascending order so that $\lambda_k \leq \lambda_{k+1}$. In particular, $|\psi_0\rangle$ is the *ground state* and λ_0 is the *ground state energy*. Without loss of generality we assume $\|H\| \leq 1$. This can always be achieved by rescaling H.

Although dynamical simulation is not the focus of this work, it is still useful to introduce the basic concepts here. The *time evolution* of the quantum system is described by the Schrodinger equation

$$i\frac{\mathrm{d}}{\mathrm{d}t}\,|\phi(t)\rangle = H\,|\phi(t)\rangle\,,$$

where $|\phi(t)\rangle$ is the state of the system. We can alternatively describe it using the *time evolution operator* e^{-iHt}, so that

$$|\phi(t)\rangle = e^{-iHt}\,|\phi(0)\rangle\,.$$

The simulation of time evolution, also known as *Hamiltonian simulation*, on a gate-based quantum computer, is a major research direction, and we will be using some of the relevant results [113, 34, 35, 115, 116, 114, 56, 34, 60, 163, 55, 153, 13, 14].

1.1 The input model

Because we want to design algorithms that run in polynomial time and space, we cannot store the value of each entry of H in classical memory, as this procedure alone would take $\mathcal{O}(\exp(n))^2$ time. In all the settings we care about in this work, e.g. quantum chemistry, condensed matter, quantum field theories, H admits a succinct classical description that takes up only $\mathrm{poly}(n)$ memory. This succinct description enables us to construct what is know as a *block encoding* of the Hamiltonian H.

[1] The overlap between two pure states $|\phi\rangle$ and $|\psi\rangle$ is defined to be $|\langle\phi|\psi\rangle|$.

[2] In this work we use the following asymptotic notations besides the usual $\mathcal{O}$ notation: we write $f = \Omega(g)$ if $g = \mathcal{O}(f)$; $f = \Theta(g)$ if $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$; $f = \widetilde{\mathcal{O}}(g)$ if $f = \mathcal{O}(g\,\mathrm{polylog}(g))$.

We say that a unitary U_H is an (α, m, ϵ)-block encoding of H if

$$\|\alpha(\langle 0^m| \otimes I_n)U_H(|0^m\rangle \otimes I_n) - H\| \leq \epsilon.$$

A more intuitive way to understand block encoding is the following: U_H is a unitary whose upper-left block is gives the Hamiltonian up to the subnormalization factor α, i.e.,

$$U_H \approx \begin{pmatrix} H/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix}.$$

In general, block encoding U_H can be implemented with poly(n) quantum gates, thus enabling us to load the Hamiltonian into our algorithm efficiently.

There are also situations in which we assume that the Hamiltonian H is accessed through its time evolution operator $e^{-i\tau H}$, where $\tau < \pi/\|H\|$ is chosen to so that the eigenvalues of H can be obtained from those of $e^{-i\tau H}$ unambiguously. Because $e^{-i\tau H}$ is already a unitary, we do not necessarily need to block encode it in another unitary.

These two input models are in fact equivalent if we ignore the overhead that comes from converting between them, which is a reasonable thing to do in the fully fault-tolerant setting. $e^{-i\tau H}$ can be obtained from the block encoding U_H through Hamiltonian simulation, and U_H can be obtained from $e^{-i\tau H}$ through matrix logarithm implemented using quantum singular value transformation [85, Corollary 71].

1.2 The ground state and the ground state energy

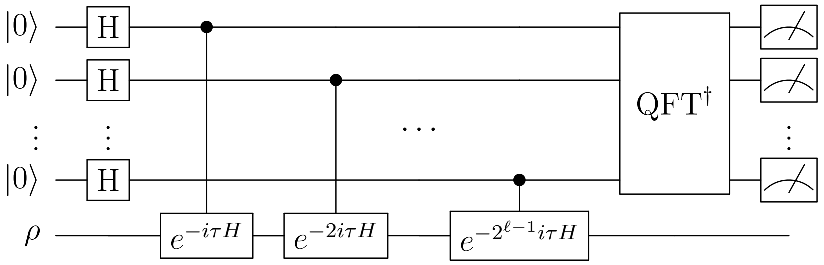
Much of this work will focus on preparing the ground state $|\psi_0\rangle$ and estimating the ground state energy λ_0 of a quantum system. To circumvent the problem presented by the QMA-completeness results, we assume that we can efficiently prepare a quantum state $|\phi_0\rangle$ such that $|\langle\psi_0|\phi_0\rangle| \geq \gamma$. γ is treated as a parameter in the cost analysis of our algorithm. If $\gamma \geq 1/\text{poly}(n)$, then on a quantum computer, with QPE, we can estimate the ground state energy λ_0 to precision $\epsilon = 1/\text{poly}(n)$ in poly(n) time.

For ground state preparation, we need a further assumption, namely that the *spectral gap*, i.e., $\lambda_1 - \lambda_0$, is at least $\Delta > 0$. If $\Delta \geq 1/\text{poly}(n)$, then the ground state and the excited states can be distinguished in polynomial time, thus enabling us to prepare the ground state in poly(n) time.

If our goal is only a polynomial time algorithm, then we can stop here. However, to actual scaling with respect to γ and ϵ matters a lot in a realistic implementation of the algorithms, since they can results in orders of magnitude difference in the runtime. A goal of this work is therefore to improve the dependence on γ and precision in the tasks of ground state energy estimation and state preparation.

1.3 The quantum phase estimation algorithm

Let us first examine in more detail how well QPE can perform the tasks. The QPE uses the following circuit:



(1.1)

In the above circuit, H denotes the Hadamard gate, and the and QFT denotes the quantum Fourier transform. There are ℓ qubits in the energy register (the qubits initialized in $|0\rangle$). τ is chosen so that $\tau\|H\| \leq \pi$ to ensure that we can determine the eigenvalues of H without worrying about whether it should be λ_k or $\lambda_k + 2\pi/\tau$. We measure the cost by how many times we need to query the time evolution operator $e^{-i\tau H}$.

Let us first have a intuitive, and idealized, understanding of what this algorithm does, and then discuss the important caveats later. After running the circuit and measuring the *energy register*, i.e., the qubits on which the time evolution $e^{-i\tau H}$ is controlled, with probability $p_k = |\langle\phi_0|\psi_k\rangle|^2$, the output will correspond to an eigenvalue λ_k, and the *state register*, i.e., the last register on which the time evolution is performed, will yield an eigenstate $|\psi_k\rangle$. Note that these probabilities add up to one because the eigenstates form a complete basis of $\mathcal{H}$ and $|\phi_0\rangle$ is a normalized eigenstate. In this way, we can simply run the QPE algorithm, and with probability p_0 we will simultaneously obtain the ground state and the ground state energy. In order to ensure that what we get is indeed the ground state rather than an excited state, we need to repeat the procedure multiple times and take the minimum energy measurement. The probability of getting λ_0 and $|\psi_0\rangle$ in this way is

$$1-(1-p_0)^{N_s}$$

if the procedure is repeated N_s times. We want this probability to be at least $1-\delta$, and consequently we need to set $N_s = \mathcal{O}(p_0^{-1}\log(\delta^{-1}))$.

The above picture is however oversimplified. It assumes that the phases $\tau\lambda_k$ can be estimated perfectly, and is therefore in violation of the Heisenberg limit [88, 87, 182, 181]. In the analysis below, we will take into account the inevitable phase error. We need to choose τ so that $\tau\lambda_0 + 2\pi > \tau\lambda_k + \tau\epsilon$. This is to ensure that the error from the largest eigenvalue does not wrap around the torus $[-\pi, \pi]$ to interfere with the lowest eigenvalue. Upon running the circuit and measuring the energy register the output will yield a phase estimate $\tau\hat{\phi}$ such that $\hat{\phi} \approx \lambda_k$, rather than $\hat{\phi} = \lambda_k$, with probability p_k. The output $\hat{\phi}$ approximates λ_k in a probabilistic sense: with probability at least $1 - \vartheta$, $|\hat{\phi} - \lambda_k| \leq \epsilon$. In other words, for each k, the event $|\hat{\phi} - \lambda_k| \leq \epsilon$ happens with probability at least $p_k(1 - \vartheta)$. In order to achieve this the number of times we need to query $e^{-i\tau H}$ is $\mathcal{O}(\vartheta^{-1}\epsilon^{-1})$.

Ground state energy estimation with QPE

We then follow the discussion in [110] to see what this means for ground state energy estimation. If we take the approach of performing QPE N_s times and taking the minimum of the energy measurements, then for an event that only has $\mathcal{O}(1/N_s)$ probability of happening in a single run, the probability of this event occurring at least once in the total N_s repetitions is now $\mathcal{O}(1)$, which means that we cannot ensure that the error happens with sufficient low probability. To ensure that the ground state energy error in the end is at most ϵ with probability at least $1 - \delta$, the error probability in each QPE run needs to be smaller than $\vartheta = \delta/N_s$. Consequently the evolution time in each run, as analyzed in the previous paragraph, is $\mathcal{O}(N_s\delta^{-1}\epsilon^{-1})$. The procedure needs to be repeated N_s times to get the ground state, and as a result the number of times we need to query $e^{-i\tau H}$ is

$$\mathcal{O}(N_s^2\delta^{-1}\epsilon^{-1}) = \mathcal{O}(p_0^{-2}\delta^{-1}\epsilon^{-1}\log^2(\delta^{-1})) = \mathcal{O}(\gamma^{-4}\delta^{-1}\epsilon^{-1}\log^2(\delta^{-1})). \tag{1.2}$$

This query complexity scaling is sub-optimal in the γ and δ dependence, making QPE expensive when the initial overlap is small and when we require high confidence in the estimate. A simple way to improve the δ dependence is to take the median classically as a post-processing step. We can set $\delta = 2/3$ in (1.2), and get multiple outputs from N_m runs. Taking the median of all these outputs ensure that the probability of having an error larger than ϵ decays exponentially, and is upper bounded by $e^{-\Omega(N_m)}$ due to the Chernoff bound. We therefore need to choose $N_m = \mathcal{O}(\log(\delta^{-1}))$ to ensure an $1 - \delta$ success probability, and the query complexity becomes

$$\mathcal{O}(\gamma^{-4}\epsilon^{-1}\log(\delta^{-1})). \tag{1.3}$$

The γ dependence is however not improved, which will be one of the focuses of this work. In fact, in Section 1.6, also discussed in more detail in Chapter 2, we will see that a quartic speedup is possible.

Ground state preparation with QPE

QPE can also help us prepare the ground state. For this task we need to assume that the spectral gap is at least $\Delta > 0$. We also need to choose τ so that $\tau\lambda_0 + 2\pi > \tau\lambda_k + \tau\Delta$ similar to what we did for energy estimation. Running the QPE circuit and before performing measurements, the quantum state of the whole system is

$$\sum_{m=0}^{2^\ell-1} |m\rangle \left(\frac{1}{2^\ell}\sum_{j=0}^{2^\ell-1} e^{ij(\frac{2m\pi}{2^\ell}-\tau H)} |\phi_0\rangle\right) = \sum_{m=0}^{2^\ell-1} |m\rangle\, K\left(\frac{2m\pi}{2^\ell} - \tau H\right) |\phi_0\rangle\,, \tag{1.4}$$

where $K(x)$ denotes the Dirichlet kernel

$$K(x) = \frac{1}{2^\ell}\sum_{j=0}^{2^\ell-1} e^{ijx}. \tag{1.5}$$

This kernel is 2π-periodic and satisfies

$$|K(x)| \leq \frac{C_K}{|2^\ell x|},$$

for $x \in [-\pi, \pi]$ and some constant C_K that does not depend on ℓ or x [126, Eq. (5.34)].

We then measure the energy register, and with probability $|\langle\phi_0|\psi_0\rangle|^2(1-\mathcal{O}(\frac{1}{2^\ell\Delta}))$ we will get a state that is $\mathcal{O}(\frac{1}{\gamma 2^\ell\Delta})$ away from $|\psi_0\rangle$ in 2-norm distance. If we want the state preparation error to be below ϵ, we need $2^\ell = \mathcal{O}(\frac{1}{\gamma\Delta\epsilon})$, and the procedure needs to be repeated $\mathcal{O}(\gamma^{-2}\log(\delta^{-1}))$ times for it to succeed with probability at least $1-\delta$. We can therefore prepare the ground state to within error ϵ using

$$\mathcal{O}(\gamma^{-3}\Delta^{-1}\epsilon^{-1}\log(\delta^{-1})) \tag{1.6}$$

queries to $e^{-i\tau H}$.

Improved implementations of QPE

The above scalings for ground state preparation (Eq. (1.6)) and energy estimation (Eq. 1.3) can be improved in various ways. The first thing we can do is to modify the

QPE kernel $K(x)$ in (1.5). This kernel decays slowly as 2^ℓ, which is proportional to the total number of queries, increases, and this results in large error in ground state preparation, and large uncertainty in ground state energy estimation. There are two methods to make the kernel decay exponentially in the number of queries. The first method is called *high-confidence QPE* [105, 135, 124], which takes multiple phase estimates and compute the median coherently (which means without measurement). In the end the coherent median will have an error probability that decays exponentially with the number of queries. In this method we need to have multiple energy registers to store the phase estimates. The second method is to utilize a resource state [26, Section II B] to implement a Kaiser window filter [145]. This method does not require multiple energy registers, and can achieve the same effect as the first method. With either of these two methods, we can solve the ground state energy estimation problem with query complexity

$$\widetilde{\mathcal{O}}(\gamma^{-2}\epsilon^{-1}\log(\delta^{-1})), \tag{1.7}$$

and the ground state preparation problem with query complexity

$$\widetilde{\mathcal{O}}(\gamma^{-2}\Delta^{-1}\log(\epsilon^{-1}\delta^{-1})). \tag{1.8}$$

For ground state preparation, if the ground state energy is known within error $< \Delta$, further improvement can be obtained by using amplitude amplification [42] instead of running the QPE multiple times. This leads to a quadratic speedup in terms of the dependence on γ. We get the following query complexity for ground state preparation

$$\widetilde{\mathcal{O}}(\gamma^{-1}\Delta^{-1}\log(\epsilon^{-1}\delta^{-1})). \tag{1.9}$$

However, knowledge of the ground state energy is necessary here because we need to know what energy measurement result to amplify. Consequently, we cannot directly apply amplitude amplification to the grounds state energy problem to get a similar quadratic speedup. Therefore, a natural question to ask is, can we estimate the ground state energy with an almost linear dependence on γ^{-1}? In Chapter 2, based on Ref. [111], we will give an affirmative answer to this question. A very important tool to solve this problem is quantum algorithms for matrix functions, which we will introduce next.

1.4 Quantum algorithms for matrix functions

For quantum algorithms for matrix functions, the goal is to get a block encoding (introduced in Section 1.1) of $f(A)$ for a function f, when A is given through its own block encoding. In this work we will focus on the case where A is Hermitian. The quantum phase estimation algorithm is closely connected to the implementation of matrix functions, as when we measure the energy register, from (1.4) we can see that we are actually implementing a matrix function $K(\frac{2m\pi}{2^\ell} - \tau H)$ if the measurement outcome is m.

In fact one of the conceptually simplest way to implement matrix functions is through QPE: we can first run QPE to get the eigenvalues of A, apply f to the eigenvalues in superposition, and apply controlled rotations to record the transformed eigenvalues in the amplitudes. This is exactly what was done in the famous Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear systems [93].

A drawback of this approach is that it is often difficult to attain high precision. The HHL algorithm solves a linear system $Ax = b$ with $\widetilde{\mathcal{O}}(\kappa^2\epsilon^{-1})$ queries to A to obtain a normalized solution. Here κ is the condition number of A. In essence, the HHL algorithm approximately implements the matrix function A^{-1} with $\widetilde{\mathcal{O}}(\kappa\epsilon^{-1})$ queries to A, and applies it to the right-hand side b, which is represented by a quantum state. To achieve the scaling mentioned above the HHL algorithm also needs to use amplitude amplification. Despite the potential exponential speedup compared to a classical linear system solver, both the dependence on κ and ϵ are sub-optimal. Two types of matrix function algorithms were later developed that enable evaluating matrix function to high precision: the linear combination of unitaries (LCU) method [34, 59] and the quantum signal processing (QSP) method [117, 115, 86]. Both methods enable us to implement the matrix function A^{-1} with $\widetilde{\mathcal{O}}(\kappa\log(\epsilon^{-1}))$ queries to A, achieving an exponential speedup in the precision dependence compared to the HHL algorithm.

If we want to use the LCU method to implement a matrix function $f(A)$, assuming A is Hermitian, we will first construct a Fourier approximation of f:

$$f(x) \approx \sum_{j=-J}^{J} C_j e^{-ij\tau x},$$

for x on certain parts of the real axis. This is important since f may not be a periodic function. The LCU circuit allows us to block encode $\sum_{j=-J}^{J} C_j e^{-ij\tau A}$, which approximates $f(A)$. Because Fourier approximation can achieve exponential accuracy for functions with good enough regularity, the LCU method can help us achieve high precision easily. The cost of this approach depends linearly on J, and also on the ℓ^1-norm of the coefficients, i.e. $\sum_j |C_j|$. Besides the ancilla qubits needed to block encode A, we need $\mathcal{O}(\log(J))$ extra ancilla qubits to store the coefficients C_j.

Rather than Fourier approximation, QSP relies on polynomial approximation, which typically yields the same error scaling as the Fourier approximation (we only

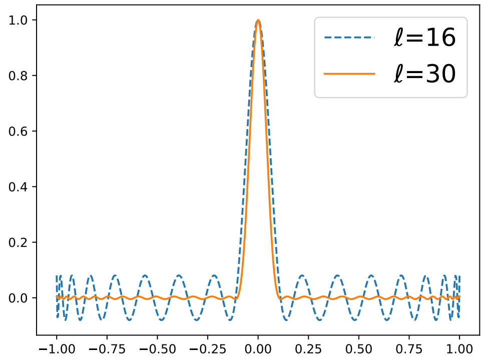


Figure 1.1: The filter polynomials with degrees 2ℓ for $\ell = 16$ and 32. The plot is from Ref. [112].

care about certain parts of the real axis, as otherwise periodicity may present a problem). Compared to LCU, QSP uses a more compact circuit, using only $\mathcal{O}(1)$ additional ancilla qubits beyond the ones required to block encode A. Moreover, it avoids the subnormalization factor that comes from summing the operators $e^{-ij\tau A}$ in LCU, and as a result can oftentimes remove some of the logarithmic factors in the complexity. QSP can also be modified into the quantum singular value transformation (QSVT) algorithm that allows implementing generalized matrix functions for non-Hermitian A. The cost of the procedure mainly depends on the degree of the polynomial. The implementation of the QSP circuit requires finding a sequence of phase factors corresponding to the polynomial we want to implement, and this can be a numerically difficult task even though it is proved to be solvable in polynomial time. Extensive work has been done in recent years on finding the phase factors in a numerically stable and efficient way [50, 70, 168, 178].

1.5 Eigenstate filtering through QSP

We will show that the matrix functions algorithms can be used to prepare eigenstates of H. Suppose we have a function $f(x)$ that is peaked at $x = 0$, with $f(0) = 0$, but falls to 0 rapidly away from $x = 0$, such as the function shown in Figure 1.1. Then we can apply it to $H - \lambda_k$, and we will get

$$f(H - \lambda_k) = |\psi_k\rangle \langle\psi_k| + \sum_{k' \neq k} f(\lambda_{k'} - \lambda_k) |\psi_{k'}\rangle \langle\psi_{k'}| \,.$$

Now suppose that there is a spectral gap Δ separating λ_k from other eigenvalues, then we can design f so that $|f(x)| \leq \epsilon$ when x is at least Δ distance away from 0 and within the range of the eigenvalues. Then we will have

$$\|f(H-\lambda_k) - |\psi_k\rangle\langle\psi_k|\,\| \leq \epsilon.$$

Therefore $f(H-\lambda_k)$ approximates the projection operator $|\psi_k\rangle\langle\psi_k|$. With this projection operator, we can apply it to a quantum state (a initial guess) that has overlap with $|\psi_k\rangle$, then the resulting quantum state will be close to $|\psi_k\rangle$. If the overlap between the initial guess and $|\psi_k\rangle$ is at least γ, then the quantum state we get through this procedure will have an $\mathcal{O}(\gamma^{-1}\epsilon)$ error.

Using the construction in Ref. [112], which is also discussed in Chapter 4, we can construct a polynomial with degree 2ℓ with $\ell = \mathcal{O}(\Delta^{-1}\log(\epsilon^{-1}))$ that satisfies the above requirements. Figure 1.1 shows the polynomial we construct. A closed-form expression can be found:

$$f(x) = \frac{T_\ell\left(-1+2\frac{x^2-\Delta^2}{1-\Delta^2}\right)}{T_\ell\left(-1+2\frac{-\Delta^2}{1-\Delta^2}\right)}, \tag{1.10}$$

where $T_\ell(x)$ is the ℓ-th Chebyshev polynomial of the first kind. This polynomial is inspired by the shifted and rescaled Chebyshev polynomial discussed in [143, Theorem 6.25]. It is in fact the optimal polynomial for this task as proved in Section 4.11 (also [112, Appendix E]).

Because the error in the polynomial construction can be amplified by a factor $\mathcal{O}(\gamma^{-1})$ in the quantum state we get, instead of precision ϵ we need to get to a higher precision $\gamma\epsilon$. Therefore the polynomial degree needed is $\mathcal{O}(\Delta^{-1}\log(\gamma^{-1}\epsilon^{-1}))$. This procedure needs to be repeated $\mathcal{O}(\gamma^{-2}\log(\delta^{-1}))$ times to ensure a success probability of at least $1-\delta$.

This automatically enables us to prepare the ground state $|\psi_0\rangle$ if we have exact knowledge of the ground state energy λ_0. However, knowing the ground state energy exactly is oftentimes too much a requirement and is in fact unnecessary.

Now suppose we know that $\lambda_0 \leq \mu - \Delta/2$ and $\lambda_1 \geq \mu + \Delta/2$ for some μ, then we can design a polynomial $f(x)$ such that, within the range of the eigenvalues of H, $|1-f(x)| \leq \epsilon$ for $x \leq -\Delta/2$, and $|f(x)| \leq \epsilon$ for $x \geq \Delta/2$. This polynomial is shaped like a step function, but instead of a jump discontinuity, it has a smooth transition from 1 to 0. Construction for such a polynomial is provided in [115, 86]. Applying this polynomial to implement $f(H-\mu)$, we will be able to prepare the ground state in much the same way as discussed above. This enables us to prepare the ground state, using amplitude amplification, with

$$\mathcal{O}(\gamma^{-1}\Delta^{-1}\log(\gamma^{-1}\epsilon^{-1})) \tag{1.11}$$

queries to H to ensure a success probability of at least $1 - \delta$. This is similar to the scaling we need for the most advanced version of QPE, and the QSP algorithm uses fewer ancilla qubits.

1.6 Binary search for ground state energy

This section briefly summarizes the binary search method, to be discussed in detail in Chapter 2, which is based on Ref. [111]. We use the matrix function methods to estimate the ground state energy. Surprisingly, this yields better query complexity then QPE. The main result is summarized in Theorem 8. We first observe that the ground state energy problem can be converted into a binary search problem. We know that the ground state energy is somewhere in the interval $[-\|H\|, \|H\|]$ ($\|H\|$ might not be known exactly but we can replace it with any upper bound of $\|H\|$). If we can determine, for any given x, whether $\lambda_0 < x$ or $\lambda_0 > x$, then we can simply do binary search to find x.

However, exactly implementing this procedure is not possible for the input model we consider due to the intrinsic uncertainty of a quantum system. What we have to settle for is a *fuzzy binary search*. At each search step, we know that $\lambda_0 \in [\lambda_L, \lambda_R]$, and we want to see which one of the following is true: either $\lambda_0 \in [\lambda_L, \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R]$, or $\lambda_0 \in [\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R, \lambda_R]$. Note that when $\lambda_0 \in [\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R, \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R]$, both of them are true. This is why we call this a fuzzy binary search. The above problem for each search step we call the *fuzzy bisection problem* [69, Definition 5]. This problem is equivalent to a *promise problem* using theoretical computer science terms: suppose we are given the promise that either $\lambda_0 \in [\lambda_L, \frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R]$ or $\lambda_0 \in [\frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R, \lambda_R]$, we are then asked to decide which one is true. The reason of this equivalence is because when $\lambda_0 \in [\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R, \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R]$ we can output anything and it is always correct for the fuzzy bisection problem we consider.

Once we can solve this fuzzy bisection problem, we can design an algorithm that proceeds iteratively. If $\lambda_0 \in [\lambda_L, \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R]$ then we can update λ_R to be $\frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R$, and if $\lambda_0 \in [\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R, \lambda_R]$ then we can update λ_L to be $\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R$. In this way we get a new pair of λ_L and λ_R such that $\lambda_L \leq \lambda_0 \leq \lambda_R$ and $\lambda_R - \lambda_L$ shrinks by 2/3. Once we have $\lambda_R - \lambda_L \leq 2\epsilon$, we can choose $\frac{\lambda_L + \lambda_R}{2}$ as our estimate for λ_0, and the estimation error will be at most ϵ.

Now we will discuss how to solve the fuzzy bisection problem using the matrix function technique. We find a polynomial f that, within the range of the eigenvalues of H, satisfies $|f(x) - 1| \leq \epsilon'$ for $x \leq \frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R$, $|f(x)| \leq \epsilon'$ for $x \geq \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R$. We shall see that the fuzzy bisection problem can be solved by looking at the amplitude $\|f(H)|\phi_0\rangle\|$ where $|\phi_0\rangle$ is the initial guess of the ground state that is guaranteed to have an overlap $|\langle\psi_0|\phi_0\rangle| \geq \gamma$. If $\lambda_0 \in [\lambda_L, \frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R]$, then

$$\|f(H)|\phi_0\rangle\| = \left\|\sum_k f(\lambda_k)\langle\psi_k|\phi_0\rangle|\psi_k\rangle\right\| \geq |f(\lambda_0)\langle\phi_0|\psi_0\rangle| \geq \gamma(1-\epsilon');$$

if $\lambda_0 \in [\frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R, \lambda_R]$, then

$$\|f(H)|\phi_0\rangle\| = \left\|\sum_k f(\lambda_k)\langle\psi_k|\phi_0\rangle|\psi_k\rangle\right\| \leq \epsilon'.$$

We choose ϵ' so that $\gamma(1-\epsilon') - \epsilon' \geq \Omega(\gamma)$. Therefore the two situations above can be distinguished by estimating the amplitude $\|f(H)|\phi_0\rangle\|$ to precision $\mathcal{O}(\gamma)$. Using the amplitude estimation algorithm [42], this can be done with $\mathcal{O}(\gamma^{-1})$ queries to $f(H)$ and the circuit preparing $|\phi_0\rangle$, to ensure a success probability of at least 2/3. The $\mathcal{O}(\gamma^{-1})$ scaling here is the key why we can estimate the ground state energy with linear dependence on γ^{-1}. Because we are solving a decision problem involving two possible outcomes, we can use majority voting to boost the success probability to $1-\delta'$ with $\mathcal{O}(\log(\delta'^{-1}))$ repetitions. There are in total $\mathcal{O}(\log(\epsilon^{-1}))$ search steps and we need all of them to succeed. Therefore in order to achieve a final success probability of $1-\delta$ we need $\delta' = \mathcal{O}(\delta\log^{-1}(\epsilon^{-1}))$.

The next key thing to figure out is the degree of f. We can use the approximate step function we discussed above for the task of ground state preparation, whose degree largely depends inverse proportionally on the length of the interval in which the function transitions from 1 to 0. In our present case, the transition interval length is $(\lambda_R - \lambda_L)/3$. For the last search step, this length is proportional to ϵ, and in all previous steps the length shrinks by 2/3 in each step. Therefore the sum of all polynomial degrees is proportional to

$$\frac{1}{\epsilon}\left(1 + \frac{2}{3} + \left(\frac{2}{3}\right)^2 + \cdots\right) = \mathcal{O}(\epsilon^{-1}).$$

Putting the above analysis together we can see that the query complexity of our algorithm is

$$\widetilde{\mathcal{O}}(\gamma^{-1}\epsilon^{-1}\log(\delta^{-1})) \tag{1.12}$$

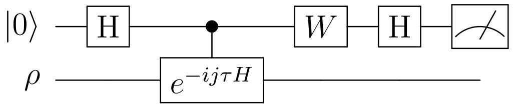
to estimate the ground state energy to within additive error ϵ with probability at least $1-\delta$, using an initial guess $|\phi_0\rangle$ with $|\langle\phi_0|\psi_0\rangle| \geq \gamma$. Thus we are able to achieve the desired quadratic improvement in terms of the γ dependence compared to QPE in 1.7.

1.7 Implementation on an early fault-tolerant quantum computer

This section is a brief summary of Chapter 3, which is based on Ref. [110]. The circuit implementation of the algorithm in the previous section uses many ancilla qubits for amplitude estimation, and requires that we need to maintain coherence, i.e., preserving the quantum state in superposition and protecting from error, during the entire computation. On early fault-tolerant quantum computers [47, 27, 41, 108], we may be able to maintain coherence for considerable length of time through quantum error correction, yet it is still desirable to reduce the coherence time requirement. More concretely, we suppose we are considering two algorithms to solve a problem. In one algorithm we need to run a circuit of depth d_1 N_1 times and take average of the results, and in the second algorithm we need to run a circuit of depth d_2 N_2 times and take average. Now we assume $d_1N_1 < d_2N_2$, and yet $d_1 > d_2$, then in the early fault-tolerant setting, it may still be preferable to use the second algorithm despite the larger total runtime. It is also preferable to reduce the number of qubits needed as much as possible, since a single logical qubit typically require a large number of physical qubit to implement for error correction.

This issue is of great importance for an early demonstration of useful quantum advantage. In order to use quantum computers to solve classically intractable problems that are at the same time useful, on the one hand we need improvement in quantum hardware and error correcting codes, and on the other hand we can design our algorithms with the these limitations in mind, and reduce the requirement on hardware. Quantum chemistry is widely considered a promising field in which we can hope to demonstrate quantum advantage, due to the computational challenges on the classical side. In this section, as well as in Chapter 3, we will design an algorithm motivated by the above discussion for the ground state energy estimation problem. Despite the limitations on the circuit depth and number of qubits, we find that we are still able to reach the Heisenberg-limited precision scaling [88, 87, 182, 181] for this problem, which is the best precision scaling possible due to limitations placed by the law of quantum mechanics.

Our algorithm uses a simple and widely used circuit of the following form



(1.13)

where H is the Hadamard gate. We choose $W = I$ or $W = S^\dagger$ where S is the phase gate, depending on the quantity we want to estimate. This is the same circuit used to perform the Hadamard test. In Ref. [110], and also in Chapter 3, we found that in fact this circuit can be used to extract spectral properties of the Hamiltonian H. In particular, it can be used to compute the cumulative distribution function (CDF) corresponding to the spectral measure

$$p(x) = \sum_{k=0}^{K-1} p_k \delta(x - \tau\lambda_k), \quad x \in [-\pi, \pi], \tag{1.14}$$

where $p_k = |\langle\phi_0|\psi_k\rangle|^2$ is the overlap between the initial guess of the ground state with the k-th eigenstate. We can extract information of the ground state energy λ_0 from the CDF, because λ_0 corresponds to the first jump discontinuity of the CDF. Using a similar binary search approach to the one we discussed in Section 1.6, we can estimate λ_0 to within additive error ϵ with total runtime that scales linearly in ϵ^{-1}. The main result is summarized in Corollary 15. The Heisenberg-limited precision scaling is reached by using randomized evolution times in the circuit, in a manner that is similar to the unbiased multi-level Monte Carlo method in [140, 141].

In [180], this framework was extended to estimate the expectation value of observables in the ground state. Ref. [166] considered using randomized compiling within this framework to reduce the circuit depth overhead that comes from implementing the time-evolution operator. More recently, in Ref. [69], it is shown that we can insert Pauli-X rotations between the controlled time evolution operators to quadratically improve the overlap dependence without increasing the circuit depth. The same paper also shows that the improvement with respect to the overlap dependence becomes quartic if one is allows a larger circuit depth. Besides the ancilla qubits and circuit depth, Ref. [8] also considered minimizing the use of multi-qubit control structures, which may be another difficulty for early fault-tolerant quantum computers. These results provide a versatile toolbox to estimate ground state properties in the early fault-tolerant setting.

1.8 Application to the quantum linear system problem

Besides the ground state problem, quantum eigenstate filtering is also useful in solving the quantum linear system problem (QLSP), which we briefly discussed in Section 1.4. This section summarizes the QLSP algorithm in Chapter 4, which is based on Ref. [112]. A natural application of the QLSP is in computing Green's function. For a Hamiltonian H, the Green's function involves quantity such as $\langle\Psi|a_i(z-H)^{-1}a_j^\dagger|\Psi\rangle$, where $|\Psi\rangle$ is the ground state, $a_i^\dagger$, a_j are the creation and annihilation operators, and

z is a complex number. This quantity naturally involves solving a linear system with coefficient matrix $z - H$, whose dimension is exponential in the system size.

In QLSP, we want to solve a linear system $Ax = b$, where A is a matrix of size $N \times N$. In the quantum setting we need to specify further what we mean here. We assume that A is given through its block encoding, and its singular values are contained in the interval $[1/\kappa, 1]$, and consequently κ is an upper bound of its condition number. Usually it is assumed that A can be constructed from the sparse matrix oracles [34, 59, 154]. b is assumed to be accessed through a unitary oracle that prepares a normalized quantum state $|b\rangle$ that is parallel to b. The goal is to prepare a normalized quantum state $|x\rangle$ that is parallel to $A^{-1}|b\rangle$. More precisely, we want a state $|x\rangle$ such that

$$\left\| |x\rangle - \frac{A^{-1}|b\rangle}{A^{-1}|b\rangle} \right\| \leq \epsilon,$$

for the allowed error $\epsilon > 0$. We are primarily interested in the query complexity, i.e., the number of queries to the above mentioned oracles. Since for all state-of-the-art algorithms the number of queries to all oracles are the same asymptotically, we do not need to specify the oracle when talking about the query complexity.

As mentioned before, the seminal HHL algorithm has a query complexity $\widetilde{\mathcal{O}}(\kappa^2\epsilon^{-1})$. This gives a potentially exponential speedup compared to classical algorithms because if A is sparse and its elements efficiently computable, then the oracles use $\mathcal{O}(\mathrm{polylog}(N))$ gates, and thus the total runtime of the algorithm is $\widetilde{\mathcal{O}}(\mathrm{polylog}(N)\kappa^2\epsilon^{-1})$. This is in contrast with classical methods that typically have $\widetilde{\mathcal{O}}(\mathrm{poly}(N, \kappa, \log(\epsilon^{-1})))$ scaling.

Compared to classical algorithms, the HHL algorithm has worse scaling with respect to κ and ϵ. Later works improved both the κ dependence and the ϵ dependence. For the κ dependence, in Ref. [12], variable-time amplitude amplification (VTAA), a generalization of the standard amplitude amplification algorithm that allows us to amplify the success probability of quantum algorithms by stopping different branches at different times, was first used to successfully improve the dependence on κ to be almost linear, and the query complexity is $\widetilde{\mathcal{O}}(\kappa\epsilon^{-3})$. In [59], combining VTAA with the LCU approach we discussed in Section 1.4 for implementing the matrix function A^{-1}, the query complexity can be improved to $\widetilde{\mathcal{O}}(\kappa\mathrm{polylog}(\epsilon^{-1})))$, which is almost optimal with respect to both κ and ϵ. A sublinear in κ scaling is not possible unless BQP = PSPACE, which is regraded as highly unlikely [93]. A similar strategy may be applied to accelerate QSVT. Despite the success in terms of the query complexity, it is worth noting that the VTAA algorithm is a complicated procedure and can be difficult to implement, and that the overhead that shows in the polylogarithmic factor can be significant.

Another approach to reach the linear in κ scaling is through casting the QLSP into an eigenvalue problem, and solving it adiabatically. In Ref. [154], the authors constructed an adiabatic path $H(s) = sH_1 + (1-s)H_0$, which adiabatically connects $|0\rangle\,|+\rangle\,|x\rangle$, an eigenstate of $H(1) = H_1$ corresponding to the eigenvalue 0, to an initial state $|0\rangle\,|-\rangle\,|b\rangle$ (a 0-eigenstate of $H(0) = H_0$) which is easy to prepare. Importantly, along the entire the adiabatic path the spectral gap of $H(s)$ is lower bounded $\Delta^*(s) = \sqrt{(1-s)^2 + s^2/\kappa^2}$ [154]. With this adiabatic path, one can either traverse it using adiabatic time evolution [15], or quantum Zeno effect with phase randomization [150, 154]. The former achieves a query complexity of $\widetilde{\mathcal{O}}(\kappa \log^4(\epsilon^{-1}))$ while the latter achieves $\widetilde{\mathcal{O}}(\kappa\epsilon^{-1})$.

In Chapter 4, based on Ref. [112], we introduce another method to traverse the adiabatic through quantum Zeno effect, replacing phase randomization with a projection operator implemented using QSP, and the filter polynomial $f(x)$ in (1.10). With this construction, we can obtain a query complexity of $\widetilde{\mathcal{O}}(\kappa \log(\epsilon^{-1}))$ for solving the QLSP. A formal statement of the result can be found in Theorem 29. This was the best query complexity for solving the QLSP until recently, when the discrete adiabatic theorem was used to design a new algorithm for this problem, yielding a $\mathcal{O}(\kappa \log(\epsilon^{-1}))$ scaling [63], which removes a $\log(\kappa)$ factor from our algorithm.

The lower bound result from Ref. [93] tells us that we cannot hope to further improve the κ dependence for generic sparse linear systems. However, for quantum systems with certain structures, it is still possible to get further improvement, even to entirely remove the condition number dependence through *preconditoning*. Ref. [161] provides such an example: for certain systems of the form $(A+B)x = b$, we can instead solve $(I + A^{-1}B)x = A^{-1}b$, and this can remove the condition number dependence in the ideal scenario, such as when we consider linear systems coming from discretizing partial differential equations.

1.9 Outlook

In this work we primarily consider algorithms based on black-box oracles, in which the structure of the Hamiltonian or the QLSP coefficient matrix is ignored. The optimality of our result, guaranteed by query complexity lower bounds, indicates that if we want to get further improvement, we need to use more structure of the problem we study. For the QLSP, the preconditioning technique provides an example of how special structure can be useful. Looking for such special structures in the ground state problem is of even greater interest due to the central role it plays in many disciplines of science.

Our ground state algorithm relies on the assumption of an initial guess with good overlap. While mean field solutions oftentimes yield surprisingly good overlap, it is very easy to construct examples in which the overlap decays exponentially with the system size [165]. On the other hand, some classical algorithms do not need such assumptions. Examples include the density matrix renormalization group algorithm (DMRG) [169, 131, 146], in which the quantum state is optimized for a growing subset of the quantum system, and density matrix embedding theory (DMET) [106, 107, 164, 44, 173, 67, 156, 68, 175, 174], in which we only need to solve small subsystems to beyond mean field accuracy because the influence of the rest of the system is incorporated into an effective Hamiltonian. The former algorithm is based on the entanglement area law [64, 71, 94, 21, 20, 17, 16, 65, 1, 2], and the latter is based on the low entanglement between the system and the environment beyond a buffer (the bath). Identifying and utilizing such structures are very important in designing quantum algorithms that overcome the limitations of the black-box oracle model mentioned above.

Designing algorithms for the early fault-tolerant setting is another direction where meaningful progress can be made. Lowering the hardware requirement for quantum computers as the error correcting technology matures is extremely important for an early demonstration of useful quantum advantage. We have discussed algorithms for ground state energy estimation for the early fault-tolerant setting in Section 1.7, but can similar things be done for other tasks, such as the QLSP and thermal state preparation? Can we design algorithms that use very few ancilla qubits and low circuit depth to perform these tasks with provable performance guarantee? These are meaningful open problems to be explored further.

Chapter 2

Near-optimal algorithms for the ground state

Preparing the ground state of a given Hamiltonian and estimating its ground energy are important but computationally hard tasks. However, given some additional information, these problems can be solved efficiently on a quantum computer. We assume that an initial state with non-trivial overlap with the ground state can be efficiently prepared, and the spectral gap between the ground energy and the first excited energy is bounded from below. With these assumptions we design an algorithm that prepares the ground state when an upper bound of the ground energy is known, whose runtime has a logarithmic dependence on the inverse error. When such an upper bound is not known, we propose a hybrid quantum-classical algorithm to estimate the ground energy, where the dependence of the number of queries to the initial state on the desired precision is exponentially improved compared to the current state-of-the-art algorithm proposed in [Ge et al. 2019]. These two algorithms can then be combined to prepare a ground state without knowing an upper bound of the ground energy. We also prove that our algorithms reach the complexity lower bounds by applying it to the unstructured search problem and the quantum approximate counting problem.

2.1 Introduction

Estimating ground energy and obtaining information on the ground state of a given quantum Hamiltonian are of immense importance in condensed matter physics, quantum chemistry, and quantum information. Classical methods suffer from the exponential growth of the size of Hilbert space, and therefore quantum computers are expected to be used to overcome this difficulty. However even for quantum computer, estimating the ground energy is a hard problem: deciding whether the smallest eigenvalue of a generic local Hamiltonian is greater than b or smaller than a for some $a < b$ is QMA-complete [102, 100, 130, 6].

Therefore to make the problem efficiently solvable we need more assumptions. We denote the Hamiltonian we are dealing with by H, and consider its spectral decomposition $H = \sum_k \lambda_k |\psi_k\rangle \langle\psi_k|$ where $\lambda_k \leq \lambda_{k+1}$. The key assumption is that we have an initial state $|\phi_0\rangle$ which can be efficiently prepared by an oracle U_I, and has some overlap with the ground state $|\psi_0\rangle$ lower bounded by γ. This is a reasonable assumption in many practical scenarios. For instance, even for strongly-correlated molecules in quantum chemistry, there is often a considerable overlap between the true ground state and the Hartree-Fock state. The latter can be trivially prepared in the molecular orbital basis, and efficiently prepared in other basis [104]. For the moment we also assume the spectral gap is bounded from below: $\lambda_1 - \lambda_0 \geq \Delta$.

With these assumptions we can already use phase estimation coupled with amplitude amplification [42] to prepare the ground state, if we further know the ground energy to high precision. To our knowledge, the most comprehensive work on ground state preparation and ground state energy estimation was done by Ge et al. [80], which provided detailed complexity estimates for well-known methods such as phase estimation, and proposed new methods to be discussed below. As analyzed in [80, Appendix A], in order to prepare the ground state to fidelity[1] $1 - \epsilon$, the runtime of the controlled-time-evolution of the Hamiltonian is $\widetilde{\mathcal{O}}(1/(\gamma^2\Delta\epsilon))$ [2], and the number of queries to U_I is $\widetilde{\mathcal{O}}(1/\gamma)$, assuming the spectral norm of H is bounded by a constant. This is however far from optimal. Poulin and Wocjan [134] proposed a method that, by executing the inverse of phase estimation to filter out the unwanted components in the initial state, can prepare a state whose energy is in a certain given range. A different choice of parameters yields a way to prepare the ground state to fidelity $1 - \epsilon$ by running the controlled-time-evolution of the Hamiltonian with $\widetilde{\mathcal{O}}(1/(\gamma\Delta)\log(1/\epsilon))$ runtime, and using $\widetilde{\mathcal{O}}(1/\gamma)$ queries to U_I [80, Appendix C].

A key difference between ground state preparation and Hamiltonian simulation, where significant progress has been made in recent years [113, 34, 35, 115, 116, 114, 56], is its non-unitary nature. The recent development of linear combination of unitaries (LCU) method [34, 59] provided a versatile tool to apply non-unitary operators. Using LCU, Ge et al. proposed a new method to filter the initial state by applying a linear combination of time-evolutions of different time length [80], which achieves the same complexity, up to logarithmic factors, as the modified version of

[1] In this work, the fidelity between states $|x\rangle, |y\rangle$ is defined to be $|\langle x|y\rangle|$.

[2] In this work the notation $\widetilde{\mathcal{O}}(f)$ means $\mathcal{O}(f \text{poly} \log(f))$ unless otherwise stated.

Poulin and Wocjan's method discussed above.

All of the above methods prepare the ground state assuming the ground energy is known to high precision. When the ground energy is unknown, Ge et al. proposed a method to estimate the ground energy using a search method called minimum label finding [80]. This method can estimate the ground energy to precision h by running the controlled-time-evolution of the Hamiltonian for $\widetilde{\mathcal{O}}(1/(\gamma h^{3/2}))$ [3], and querying U_I $\widetilde{\mathcal{O}}(1/(\gamma\sqrt{h}))$ times. It is worth noting that their method requires $h = \widetilde{\mathcal{O}}(\Delta)$, and therefore is very expensive when the gap is extremely small. When the ground energy is not known a priori, Ge et al. proposed a method to first estimate the ground energy and then apply the LCU approach.

In recent years several hybrid quantum-classical algorithms have been developed to estimate the ground energy, or to prepare the ground state, or both. The variational quantum eigenvalue solver (VQE) [133] has gained much attention recently because of its low requirement for circuit depth and its variational structure. However the exact complexity of this algorithm is not clear because it relies on a proper choice of ansatz and needs to solve a non-convex optimization problem. Other such algorithms include quantum imaginary-time evolution, quantum Lanczos [123], and quantum filter diagonalization [132, 151]. Their complexities are either quasi-polynomial or unknown.

The recent development of block-encoding [34] and quantum signal processing (QSP) [117, 115, 86] enables us to apply non-unitary operators, specifically polynomials of a block-encoded matrix efficiently. It uses a minimal number of ancilla qubits, and avoids the Hamiltonian simulation. These will be the basic tools of this work, of which we give a brief introduction below.

Block-encoding is a powerful tool to represent a non-unitary matrix in the quantum circuit. A matrix $A \in \mathbb{C}^{N\times N}$ where $N = 2^n$ can be encoded in the upper-left corner of an $(m+n)$-qubit unitary matrix if

$$\|A - \alpha(\langle 0^m| \otimes I)U(|0^m\rangle \otimes I)\|_2 \leq \epsilon. \tag{2.1}$$

In this case we say U is an (α, m, ϵ)-block-encoding of A. Many matrices of practical interests can be efficiently block-encoded. In particular we will discuss the block-encoding of Hamiltonians of physical systems in Section 2.7.

Using the block-encoding of a Hermitian A, QSP enables us to construct block-encodings for a large class of polynomial eigenvalue transformations of A. We pay

[3] In [80], the meaning of the notation $\widetilde{\mathcal{O}}(\cdot)$ is different from that in our work. In particular, $\widetilde{\mathcal{O}}(\cdot)$ in [80] hides all factors that are poly-logarithmic in $1/h$, $1/\epsilon$, $1/\gamma$, and $1/\Delta$, regardless of what is inside the parentheses. We preserve their notation when citing their results since these factors do not play an important role when comparing the complexities of our methods.

special attention to even or odd polynomials with real coefficients, because we only apply this type of polynomial eigenvalue transformation in this work. Also for simplicity we assume the block-encoding is done without error. [86, Theorem 2] enables us to perform eigenvalue transformation of A for polynomials of definite parity (even or odd).

Theorem 1 (QSP for polynomials of definite parity)**.** *Let U be an $(\alpha, m, 0)$-block-encoding of a Hermitian matrix A. Let $P \in \mathbb{R}[x]$ be a degree-ℓ even or odd real polynomial and $|P(x)| \leq 1$ for any $x \in [-1, 1]$. Then there exists an $(1, m+1, 0)$-block-encoding $\widetilde{U}$ of $P(A/\alpha)$ using ℓ queries of U, $U^\dagger$, and $\mathcal{O}((m+1)\ell)$ other primitive quantum gates.*

Remark 2. *[86, Theorem 2] provides a singular value transformation for any square matrix A and polynomials of definite parity. When A is a Hermitian matrix, the eigenvalue transformation is the same as the singular value transformation [86, Page 203]. A related statement in the same paper is [86, Theorem 31], which describes the eigenvalue transformation of a Hermitian matrix for an arbitrary polynomial, by means of a linear combination of two polynomials of even and odd parities respectively.*

Constructing the quantum circuit for QSP requires computing a sequence of phase factors beforehand, and there are classical algorithms capable of doing this [92]. Some recent progress has been made to efficiently compute phase factors for high-degree polynomials to high precision [50, 70]. In this work, unless otherwise specified, we assume the phase factors are computed without error.

Using the tools introduced above, we assume the Hamiltonian H is given in its $(\alpha, m, 0)$-block-encoding U_H. This, together with U_I, are the two oracles we assume we are given in this work. QSP enables us to filter eigenstates using fewer qubits than LCU. In [112] a filtering method named optimal eigenstate filtering is introduced. It is based on an explicitly constructed optimal minimax polynomial, and achieves the same asymptotic complexity, ignoring poly-logarithmic factors, as the method by Ge et al. when applied to the ground state preparation problem if the ground energy is known exactly.

In this work we first develop a filtering method that filters out all eigenstates corresponding to eigenvalues above a certain threshold. This filtering method enables us to prepare the ground state of a Hamiltonian with spectral gap bounded away from zero when only an upper bound of the ground energy is known, unlike in the filtering methods discussed above which all require either exact value or high-precision estimate of the ground energy. Our filtering method has an exponentially improved dependence on precision compared to Kitaev's phase estimation [101] and uses fewer qubits compared to other variants of the phase estimation algorithm [134, 80]. This filtering method, applied to the initial state given in our assumption, also enables us to tell whether the ground energy is smaller than a or greater than b for some $b > a$, with high probability. Therefore a binary search yields a ground energy estimate with success probability arbitrarily close to one. We then combine the filtering method and ground energy estimation to prepare the ground state when no non-trivial bound for the ground energy is known. A comparison of the query complexities between the method in our work and the corresponding ones in [80], which to our best knowledge achieve state-or-the-art query complexities, are shown in Table 2.1.

		Preparation (bound known)	Ground energy	Preparation (bound unknown)
U_H	This work	$\mathcal{O}\left(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\epsilon})\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma h}\log(\frac{1}{\vartheta})\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\vartheta\epsilon})\right)$
	Ge et al.	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma\Delta}\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha^{3/2}}{\gamma h^{3/2}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha^{3/2}}{\gamma\Delta^{3/2}}\right)$
U_I	This work	$\mathcal{O}\left(\frac{1}{\gamma}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\log(\frac{\alpha}{h})\log(\frac{1}{\vartheta})\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\log(\frac{\alpha}{\Delta})\log(\frac{1}{\vartheta})\right)$
	Ge et al.	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\sqrt{\frac{\alpha}{h}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\sqrt{\frac{\alpha}{\Delta}}\right)$
Extra	This work	$\mathcal{O}(1)$	$\mathcal{O}(\log(\frac{1}{\gamma}))$	$\mathcal{O}(\log(\frac{1}{\gamma}))$
qubits	Ge et al.	$\mathcal{O}(\log(\frac{1}{\Delta}\log(\frac{1}{\epsilon})))$	$\mathcal{O}(\log(\frac{1}{h}))$	$\mathcal{O}(\log(\frac{1}{\Delta}\log(\frac{1}{\epsilon})))$

Table 2.1: The query complexities of algorithms and number of extra qubits used in our work and the corresponding ones by Ge et al. in [80]. $\alpha, \gamma, \Delta, \epsilon$ are the same as above and h is the precision of the ground energy estimate. By extra qubits we mean the ancilla qubits that are not part of the block-encoding. In this work the ground energy estimation algorithm and the algorithm to prepare ground state without a priori bound have success probabilities lower bounded by $1 - \vartheta$, while in [80] the corresponding algorithms have constant success probabilities. The complexities for algorithms by Ge et al. are estimated assuming Hamiltonian simulation is done as in [114]. The usage of the notation $\widetilde{\mathcal{O}}$ is [80] different from that in our work, as explained in footnote 3.

From the query complexities in Table 2.1 we can see our method for ground energy estimation achieves a exponential speedup in terms of the dependence of number of queries to U_I on the ground energy estimate precision h and a speedup of $1/\sqrt{h}$ factor in the dependence of number of queries to U_H on the precision. Moreover, Ge et al. assumes in their work that the precision $h = \widetilde{\mathcal{O}}(\Delta)$, while we make no such assumptions. This gives our algorithm even greater advantage when the gap is much smaller than desired precision. This becomes useful in the case of preparing a low energy state (not necessarily a ground state). Because Ge et al. used a slightly different query assumption, i.e. access to time-evolution rather than block-encoding, when computing the complexities for methods in [80] in Table 2.1 we assume the Hamiltonian simulation is done with $\mathcal{O}(\alpha t)$ queries to U_H, and the error is negligible. This can be achieved using the Hamiltonian simulation in [114], and cannot be asymptotically improved because of the complexity lower bound proved in [34]. Therefore the comparison here is fair even though our work makes use of a different oracle. Also [80] assumed a scaled Hamiltonian H with its spectrum contained in $[0, 1]$. We do not make such an assumption, and therefore the α factor should be properly taken into account as is done in Table 2.1.

Organization: The rest of the paper is organized as follows. In Section 2.2 we use QSP to construct block-encodings of reflectors and projectors associated with eigen-subspaces. In Section 2.3 we use the projectors to prepare ground state when an upper bound of the ground energy is given. In Section 2.4 we introduce the ground energy estimation algorithm, a hybrid quantum-classical algorithm based on the binary search, and use it to prepare the ground state when no ground energy upper bound is known a priori. In Section 2.5 we show the dependence of our query complexities on the overlap and gap is essentially optimal by considering the unstructured search problem. We also show the dependence of our ground energy estimation algorithm on the precision is nearly optimal by considering the quantum approximate counting problem. In Section 2.6 we use our methods to prepare low-energy states when the spectral lower gap is unknown, or even when the ground state is degenerate. In Section 2.7 we discuss practical issues and future research directions.

2.2 Block-encoding of reflector and projector

A key component in our method is a polynomial approximation of the sign function in the domain $[-1, -\delta] \cup [\delta, 1]$. The error scaling of the best polynomial approximation has been studied in [75], and an explicit construction of a polynomial with the same error scaling is provided in [115] based on the approximation of the erf function. We quote [86, Lemma 14] here with some small modification:

Lemma 3 (Polynomial approximation of the sign function)**.** *For all* $0 < \delta < 1$*,* $0 < \epsilon < 1$*, there exists an efficiently computable odd polynomial* $S(\cdot;\delta,\epsilon) \in \mathbb{R}[x]$ *of degree* $\ell = \mathcal{O}(\frac{1}{\delta}\log(\frac{1}{\epsilon}))$*, such that*

(1) for all $x \in [-1,1]$*,* $|S(x;\delta,\epsilon)| \leq 1$*, and*

(2) for all $x \in [-1,-\delta] \cup [\delta,1]$*,* $|S(x;\delta,\epsilon) - \operatorname{sign}(x)| \leq \epsilon$*.*

Remark 4. *Compared to [86, Lemma 14] we have rescaled the interval from* $[-2,2]$ *to* $[-1,1]$*, and this does not result in any substantial change.*

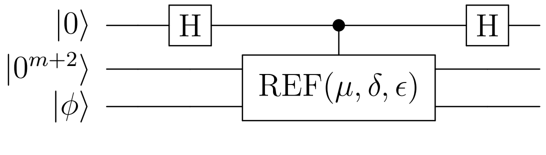
When we have the $(\alpha, m, 0)$-block-encoding of a Hermitian matrix $H = \sum_k \lambda_k |\psi_k\rangle \langle\psi_k| \in \mathbb{C}^{N\times N}$, $N = 2^n$, $\lambda_k \leq \lambda_{k+1}$, we can construct a $(\alpha + |\mu|, m+1, 0)$-block-encoding of matrix $H - \mu I$ using of [86, Lemma 29] for any $\mu \in \mathbb{R}$. Then using QSP, by Theorem 1, we can obtain an $(1, m+2, 0)$-block-encoding of $-S(\frac{H-\mu I}{\alpha+|\mu|};\delta,\epsilon)$ for any δ and ϵ. If we assume further that $\Delta/2 \leq \min_k |\mu - \lambda_k|$, then we let $\delta = \frac{\Delta}{4\alpha}$, and by Lemma 3 all the eigenvalues of $-S(\frac{H-\mu I}{\alpha+|\mu|};\delta,\epsilon)$ are ϵ-close to either 0 or 1. Therefore $-S(\frac{H-\mu I}{\alpha+|\mu|};\delta,\epsilon)$ is ϵ-close, in operator norm, to the reflector about the direct sum of eigen-subspaces corresponding to eigenvalues smaller than μ:

$$R_{<\mu} = \sum_{k:\lambda_k<\mu} |\psi_k\rangle \langle\psi_k| - \sum_{k:\lambda_k>\mu} |\psi_k\rangle \langle\psi_k|,$$

and thus the block-encoding is also an $(1, m+2, \epsilon)$-block-encoding of $R_{<\mu}$. We denote this block-encoding by $\mathrm{REF}(\mu,\delta,\epsilon)$. We omitted the dependence on H because H as well as its block-encoding is usually fixed in the rest of the paper.

In the above discussion we have used QSP in a black-box manner. For concreteness, we present a single-qubit illustrative example to demonstrate how to use a block-encoded Hamiltonian to construct the reflector in Appendix 2.8.

Because our goal is to prepare the ground state, we will use the projector more often than the reflector. Now we construct a block-encoding of projector using $\mathrm{REF}(\mu,\delta,\epsilon)$ by the following circuit



(2.2)

where H is the Hadamard gate, and we denote this circuit as $\mathrm{PROJ}(\mu,\delta,\epsilon)$. Note that

$$\begin{aligned}&(\langle 0^{m+3}|\otimes I)\mathrm{PROJ}(\mu,\delta,\epsilon)(|0^{m+3}\rangle\otimes I)\\&=\Big(\langle +|\langle 0^{m+2}|\otimes I\Big)\Big(|0\rangle\langle 0|\otimes I\otimes I+|1\rangle\langle 1|\otimes\mathrm{REF}(\mu,\delta,\epsilon)\Big)\Big(|+\rangle|0^{m+2}\rangle\otimes I\Big)\\&=\frac{1}{2}\Big(I+(\langle 0^{m+2}|\otimes I)\mathrm{REF}(\mu,\delta,\epsilon)(|0^{m+2}\rangle\otimes I)\Big),\end{aligned}$$

and we have

$$\begin{aligned}&\|(\langle 0^{m+3}|\otimes I)\mathrm{PROJ}(\mu,\delta,\epsilon)(|0^{m+3}\rangle\otimes I)-P_{<\mu}\|\\&\leq\frac{1}{2}\|(\langle 0^{m+2}|\otimes I)\mathrm{REF}(\mu,\delta,\epsilon)(|0^{m+2}\rangle\otimes I)-R_{<\mu}\|\\&\leq\frac{\epsilon}{2}.\end{aligned}$$

Here $P_{<\mu}$ is the projector into the direct sum of eigen-subspaces corresponding to eigenvalues smaller than μ

$$P_{<\mu}=\sum_{k:\lambda_k<\mu}|\psi_k\rangle\langle\psi_k|=\frac{1}{2}(I+R_{<\mu}).$$

Therefore $\mathrm{PROJ}(\mu,\delta,\epsilon)$ is an $(1,m+3,\epsilon/2)$-block-encoding of $P_{<\mu}$. In fact this can still be seen as an application of linear combination of block encoding [86, Lemma 29], using the relation $P_{<\mu}=\frac{1}{2}(R_{<\mu}+I)$.

We use the following lemma to summarize the results

Lemma 5 (Reflector and projector)**.** *Given a Hermitian matrix H with its $(\alpha,m,0)$-block-encoding U_H, with the guarantee that $\mu\in\mathbb{R}$ is separated from the spectrum of H by a gap of at least $\Delta/2$, we can construct an $(1,m+2,\epsilon)$-block-encoding of $R_{<\mu}$, and an $(1,m+3,\epsilon/2)$-block-encoding of $P_{<\mu}$, both using $\mathcal{O}(\frac{\alpha}{\Delta}\log(\frac{1}{\epsilon}))$ applications of U_H and $U_H^\dagger$, and $\mathcal{O}(\frac{m\alpha}{\Delta}\log(\frac{1}{\epsilon}))$ other one- and two-qubit gates.*

We remark that for the block-encoding $\mathrm{PROJ}(\mu,\delta,\epsilon)$, even a failed application of it can give us potentially useful information. We have

$$\mathrm{PROJ}(\mu,\delta,\epsilon)|0^{m+3}\rangle|\phi\rangle=|0\rangle|0^{m+2}\rangle P_{<\mu}|\phi\rangle+|1\rangle|0^{m+2}\rangle P_{>\mu}|\phi\rangle+\frac{1}{\sqrt{2}}|-\rangle|E\rangle,$$

where $P_{>\mu}=I-P_{<\mu}$ and $|E\rangle$ satisfies $\||E\rangle\|\leq\epsilon$. Thus when we apply the block-encoding and measure the first two registers, i.e. the first $m+3$ qubits, we have probability at least $1-\frac{\epsilon^2}{2}$ to obtain an outcome with either 0 or 1 followed by $(m+2)$ 0's. In the former case the projection has been successful, and in the latter case we have obtained an approximation of $P_{>\mu}\left|\phi\right\rangle$.

If we do not treat the output of 1 followed by $m+2$ 0's as failure then there is another interpretation of the circuit $\mathrm{PROJ}(\mu,\delta,\epsilon)$: this is an approximate projective measurement $\{P_{<\mu},P_{>\mu}\}$. In fact the whole circuit can be seen as phase estimation on a reflector, which needs only one ancilla qubit.

2.3 Algorithm with a priori ground energy bound

With the approximate projector developed in the previous section we can readily design an algorithm to prepare the ground state. We assume we have the Hamiltonian H given through its block-encoding as in the last section. If we are further given an initial state $\left|\phi_0\right\rangle$ prepared by a unitary U_I, i.e. $U_I\left|0^n\right\rangle=\left|\phi_0\right\rangle$, and the promises that for some known $\gamma>0$, μ, and Δ, we have

(P1) Lower bound for the overlap: $|\left\langle\phi_0|\psi_0\right\rangle|\geq\gamma$,

(P2) Bounds for the ground energy and spectral gap: $\lambda_0\leq\mu-\Delta/2<\mu+\Delta/2\leq\lambda_1$.

Here μ is an upper bound for the ground energy, Δ is a lower bound for the spectral gap, and γ is a lower bound for the initial overlap. Now suppose we want to prepare the ground state to precision ϵ, we can use Lemma 5 to build a block-encoding of the projector $P_{<\mu}=\left|\psi_0\right\rangle\left\langle\psi_0\right|$, and then apply it to $\left|\phi_0\right\rangle$ which we can prepare. This will give us something close to $\left|\psi_0\right\rangle$. We use fidelity to measure how close we can get. To achieve $1-\epsilon$ fidelity we need to use circuit $\mathrm{PROJ}(\mu,\Delta/4\alpha,\gamma\epsilon)$, and we denote,

$$\widetilde{P}_{<\mu}=(\langle 0^{m+3}|\otimes I)\mathrm{PROJ}(\mu,\Delta/4\alpha,\gamma\epsilon)(|0^{m+3}\rangle\otimes I)$$

then the resulting fidelity will be

$$\frac{|\left\langle\psi_0|\widetilde{P}_{<\mu}|\phi_0\right\rangle|}{\|\widetilde{P}_{<\mu}\left|\phi_0\right\rangle\|}\geq\frac{|\left\langle\psi_0|\phi_0\right\rangle|-\gamma\epsilon/2}{|\left\langle\psi_0|\phi_0\right\rangle|+\gamma\epsilon/2}\geq 1-\frac{\gamma\epsilon}{|\left\langle\psi_0|\phi_0\right\rangle|}\geq 1-\epsilon.$$

Here we have used

$$\|\widetilde{P}_{<\mu}\left|\phi_0\right\rangle\|\leq\|P_{<\mu}\left|\phi_0\right\rangle+(\widetilde{P}_{<\mu}-P_{<\mu})\left|\phi_0\right\rangle\|\leq|\left\langle\psi_0|\phi_0\right\rangle|+\gamma\epsilon/2.$$

This is when we have a successful application of the block-encoding. The success probability is

$$\|\widetilde{P}_{<\mu}|\phi_0\rangle\|^2 \geq \left(\|P_{<\mu}|\phi_0\rangle\| - \frac{\gamma\epsilon}{2}\right)^2 \geq \gamma^2\left(1-\frac{\epsilon}{2}\right)^2.$$

With amplitude amplification [42] we can boost the success probability to $\Omega(1)$ with $\mathcal{O}(\frac{1}{\gamma})$ applications of $\mathrm{PROJ}(\mu, \Delta/4\alpha, \gamma\epsilon)$ and its inverse, as well as $\mathcal{O}(\frac{m}{\gamma})$ other one- and two- qubit gates. Here we are describing the expected complexity since the procedure succeeds with some constant probability. In amplitude amplification we need to use a reflector similar to the oracle used in Grover's search algorithm [90]. Instead of constructing a reflector from $\mathrm{PROJ}(\mu, \Delta/4\alpha, \gamma\epsilon)$ we can directly use $\mathrm{REF}(\mu, \Delta/4\alpha, \gamma\epsilon)$ constructed in the previous section.

We summarize the results in the following theorem

Theorem 6 (Ground state preparation with a priori ground energy bound)**.** *Suppose we have Hamiltonian $H = \sum_k \lambda_k |\psi_k\rangle\langle\psi_k| \in \mathbb{C}^{N\times N}$, where $\lambda_k \leq \lambda_{k+1}$, given through its $(\alpha, m, 0)$-block-encoding U_H. Also suppose we have an initial state $|\phi_0\rangle$ prepared by circuit U_I, as well as the promises* (P1) *and* (P2)*. Then the ground state $|\psi_0\rangle$ can be prepared to fidelity $1-\epsilon$ with the following costs:*

1. *Query complexity: $\mathcal{O}(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\gamma\epsilon}))$ queries to U_H and $\mathcal{O}(\frac{1}{\gamma})$ queries to U_I,*

2. *Number of qubits: $\mathcal{O}(n+m)$,*

3. *Other one- and two- qubit gates: $\mathcal{O}(\frac{m\alpha}{\gamma\Delta}\log(\frac{1}{\gamma\epsilon}))$.*

2.4 Algorithm without a priori ground energy bound

Next we consider the case when we are not given a known μ to bound the ground energy from above. All other assumptions about H and its eigenvalues and eigenstates are identical to the previous sections. The basic idea is to test different values for μ and perform a binary search. This leads to a quantum-classical hybrid method that can estimate the ground energy as well as preparing the ground state to high precision.

All eigenvalues must be in the interval $[-\alpha, \alpha]$, thus we first partition $[-\alpha, \alpha]$ by grid points $-\alpha = x_0 < x_1 < \ldots < x_G = \alpha$, where $x_{k+1} - x_k = h$ for all k. Then we attempt to locate λ_0 in a small interval between two grid points (not necessarily adjacent, but close) through a binary search. To do a binary search we need to be able to tell whether a given x_k is located to the left or right of λ_0. Because of the random nature of measurement we can only do so correctly with some probability, and we want to make this probability as close to 1 as possible. This is achieved using a technique we call binary amplitude estimation.

Lemma 7 (Binary amplitude estimation)**.** *Let U be a unitary that acts on two registers, the first register indicating success or failure. Let $A = \|(\langle 0| \otimes I)U(|0\rangle\,|0\rangle)\|$ be the success amplitude. Given γ_0 and γ_1, $\Delta := \gamma_1 - \gamma_0 > 0$, provided that A is either smaller than γ_0 or greater than γ_1, we can correctly distinguish between the two cases, i.e. output 0 for the former and 1 for the latter, with probability $1-\delta$ using $\mathcal{O}((1/\Delta)\log(1/\delta))$ applications of (controlled-) U and its inverse.*

Proof. The proof is essentially identical to the proof for gapped phase estimation in [12, 59]. We can perform amplitude estimation up to error $\Delta/4$ with $\mathcal{O}(1/\Delta)$ applications of U and $U^\dagger$. This has a success probability of $8/\pi^2$ according to Theorem 12 of [42]. We turn the estimation result into a boolean indicating whether it is larger or smaller than $(\gamma_0+\gamma_1)/2$. The boolean is correct with probability at least $8/\pi^2$. Then we do a majority voting to boost this probability. Chernoff bound guarantees that to obtain a $1-\delta$ probability of getting the correct output we need to repeat $\mathcal{O}(\log(1/\delta))$ times. Therefore in total we need to run U and $U^\dagger$ $\mathcal{O}((1/\Delta)\log(1/\delta))$ times. □

We then apply binary amplitude estimation to the block-encoding of the projector defined in (2.2) $\mathrm{PROJ}(x_k, h/2\alpha, \epsilon')$ for some precision ϵ' to be chosen. We denote the amplitude of the "good" component after applying block-encoding by

$$A_k = \|(\langle 0^{m+3}| \otimes I)\mathrm{PROJ}(x_k, h/2\alpha, \epsilon')(|0^{m+3}\rangle\,|\phi\rangle)\|,$$

which satisfies the following:

$$A_k \begin{cases} \geq \gamma - \frac{\epsilon'}{2}, & \lambda_0 \leq x_{k-1}, \\ \leq \frac{\epsilon'}{2}, & \lambda_0 \geq x_{k+1}. \end{cases}$$

We can then let

$$\epsilon' = \gamma/2,$$

the two amplitudes are separated by a gap lower bounded by $\gamma/2$. Therefore we can run the binary amplitude estimation, letting U in Lemma 7 be

$$U = \mathrm{PROJ}(x_k, h/2\alpha, \epsilon')(I \otimes U_I),$$

to correctly distinguish the two cases where $\lambda_0 \leq x_{k-1}$ and $\lambda_0 \geq x_{k+1}$ with probability $1-\delta$, by running $\mathrm{PROJ}(x_k, h/2\alpha, \epsilon')$, U_I, and their inverses $\mathcal{O}((1/\gamma)\log(1/\delta))$ times. The output of the binary amplitude estimation is denoted by B_k.

We then define $\mathcal{E}$ as the event that an error occurs in the final result of binary amplitude estimation when we are computing B_k for some k such that $x_{k+1} < \lambda_0$ or $x_{k-1} > \lambda_0$ in our search process. All future discussion is conditional on $\mathcal{E}^c$ meaning that there is no error in binary amplitude estimation for B_k when $x_{k+1} < \lambda_0$ or $x_{k-1} > \lambda_0$. This has a probability that is at least $(1-\delta)^R$ where R is the number of times binary amplitude estimation is run.

Conditional on $\mathcal{E}^c$, almost surely (with probability 1) $B_k = 1$ when $\lambda_0 \leq x_{k-1}$ and $B_k = 0$ when $\lambda_0 \geq x_{k+1}$. Therefore $B_k = 0$ tells us $\lambda_0 > x_{k-1}$ and $B_k = 1$ tells us $\lambda_0 < x_{k+1}$. B_k and B_{k+1} combined give us the information as shown in Table 2.2.

B_k	B_{k+1}	Position of λ_0
1	1	$\lambda_0 < x_{k+1}$
0	0	$\lambda_0 > x_k$
0	1	$x_{k-1} < \lambda_0 < x_{k+2}$
1	0	$x_k < \lambda_0 < x_{k+1}$

Table 2.2: Conditional on $\mathcal{E}^c$, B_k and B_{k+1} can provide us with the information as shown in the table.

Algorithm 1 Binary search to locate λ_0

```
$L \leftarrow 0$, $U \leftarrow G$
while $U - L > 3$ do
    $k = \lfloor (L+U)/2 \rfloor$
    Run binary amplitude estimation to get $B_k$ and $B_{k+1}$.
    switch $(B_k, B_{k+1})$
        case $(1,1)$: $U \leftarrow k+1$
        case $(0,0)$: $L \leftarrow k$
        case $(0,1)$: return $k-1$, $k+2$
        case $(1,0)$: return $k$, $k+1$
    end switch
end while
return $L$, $U$
```

Using the Table 2.2 we can do the binary search as outlined in Algorithm 1. For the ℓ-th step in Algorithm 1 we denote the integer variables U and L by U_ℓ and L_ℓ. In all four outcomes for (B_k, B_{k+1}), if the algorithm does not terminate at this step, then the new $U_{\ell+1} - L_{\ell+1}$ will be at most $(U_\ell - L_\ell)/2 + 1$. Since $U_0 - L_0 = G$ at the very beginning, we can show inductively $U_\ell - L_\ell \leq (G-2)/2^\ell + 2$. Therefore when $\ell \geq \log_2(G-2)$ we have $U_\ell - L_\ell \leq 3$. Thus the algorithm must terminate in $\lceil \log_2(G) \rceil = \mathcal{O}(\log(\alpha/h))$ steps. The output we denote by L and U. They satisfy $x_L < \lambda_0 < x_U$ and $U - L \leq 3$.

If we want the whole procedure to be successful with probability at least $1-\vartheta$, then we need $\mathrm{Prob}(\mathcal{E}^c) \geq 1 - \vartheta$. Since

$$\mathrm{Prob}(\mathcal{E}^c) \geq (1-\delta)^{\lceil \log_2(G) \rceil} \geq (1-\delta)^{\log_2(4\alpha/h)},$$

we only need, for small ϑ,

$$\delta \leq \frac{\vartheta}{2\log_2(4\alpha/h)}.$$

Algorithm 1 enables us to locate λ_0 within an interval of length at most $3h$. In total we need to run binary amplitude estimation at most $\mathcal{O}(\log(\alpha/h))$ times. Each amplitude estimation queries $\mathrm{PROJ}(x_k, h/2\alpha, \epsilon')$ and U_I $\mathcal{O}((1/\gamma)\log(1/\delta))$ times, where $\epsilon' = \gamma/2$. Therefore the number of queries to U_H and U_I are respectively

$$\mathcal{O}\left(\frac{\alpha}{\gamma h}\log\left(\frac{\alpha}{h}\right)\log\left(\frac{1}{\gamma}\right)\log\left(\frac{\log(\alpha/h)}{\vartheta}\right)\right), \quad \mathcal{O}\left(\frac{1}{\gamma}\log\left(\frac{\alpha}{h}\right)\log\left(\frac{\log(\alpha/h)}{\vartheta}\right)\right).$$

In particular, in the procedure above we did not use (P2) but only used (P1). Therefore we do not need to assume the presence of a gap. The result can be summarized into the following theorem:

Theorem 8 (Ground energy)**.** *Suppose we have Hamiltonian $H = \sum_k \lambda_k |\psi_k\rangle\langle\psi_k| \in \mathbb{C}^{N\times N}$, where $\lambda_k \leq \lambda_{k+1}$, given through its $(\alpha, m, 0)$-block-encoding U_H. Also suppose we have an initial state $|\phi_0\rangle$ prepared by circuit U_I, as well as the promise* (P1)*. Then the ground energy can be estimated to precision h with probability $1-\vartheta$ with the following costs:*

1. *Query complexity: $\mathcal{O}\left(\frac{\alpha}{\gamma h}\log\left(\frac{\alpha}{h}\right)\log\left(\frac{1}{\gamma}\right)\log\left(\frac{\log(\alpha/h)}{\vartheta}\right)\right)$ queries to U_H and $\mathcal{O}\left(\frac{1}{\gamma}\log\left(\frac{\alpha}{h}\right)\log\left(\frac{\log(\alpha/h)}{\vartheta}\right)\right)$ queries to U_I,*
2. *Number of qubits: $\mathcal{O}(n + m + \log(\frac{1}{\gamma}))$,*
3. *Other one- and two- qubit gates: $\mathcal{O}\left(\frac{m\alpha}{\gamma h}\log\left(\frac{\alpha}{h}\right)\log\left(\frac{1}{\gamma}\right)\log\left(\frac{\log(\alpha/h)}{\vartheta}\right)\right)$.*

The extra $\mathcal{O}(\log(1/\gamma))$ qubits needed come from amplitude estimation, which uses phase estimation. If we use Kitaev's original version of phase estimation using only a single qubit [101], we can reduce the number of extra qubits to $\mathcal{O}(1)$. With Theorem 8 we can then use Algorithm 1 to prepare the ground state without knowing an upper bound for the ground energy beforehand, when in addition to (P1) we have a lower bound for the spectral gap:

(P2') Bound for the spectral gap: $\lambda_1 - \lambda_0 \geq \Delta$.

We first run Algorithm 1 to locate the ground energy in an interval $[x_L, x_U]$ of length at most Δ. Then we simply apply $\mathrm{PROJ}((x_L + x_U)/2, \Delta/4\alpha, \gamma\epsilon)$ to $|\phi_0\rangle$. This will give us an approximate ground state with at least $1-\epsilon$ fidelity. Therefore we have the following corollary:

Corollary 9 (Ground state preparation without a priori bound)**.** *Suppose we have Hamiltonian $H = \sum_k \lambda_k |\psi_k\rangle\langle\psi_k| \in \mathbb{C}^{N\times N}$, where $\lambda_k \leq \lambda_{k+1}$, given through its $(\alpha, m, 0)$-block-encoding U_H. Also suppose we have an initial state $|\phi_0\rangle$ prepared by circuit U_I, as well as the promises* (P1) *and* (P2')*. Then the ground state can be can be prepared to fidelity $1-\epsilon$ with probability $1-\vartheta$ with the following costs:*

1. *Query complexity: $\mathcal{O}\left(\frac{\alpha}{\gamma\Delta}\left(\log\left(\frac{\alpha}{\Delta}\right)\log\left(\frac{1}{\gamma}\right)\log\left(\frac{\log(\alpha/\Delta)}{\vartheta}\right)+\log\left(\frac{1}{\epsilon}\right)\right)\right)$ queries to U_H and $\mathcal{O}\left(\frac{1}{\gamma}\log\left(\frac{\alpha}{\Delta}\right)\log\left(\frac{\log(\alpha/\Delta)}{\vartheta}\right)\right)$ queries to U_I,*
2. *Number of qubits: $\mathcal{O}(n+m+\log(\frac{1}{\gamma}))$,*
3. *Other one- and two- qubit gates: $\mathcal{O}\left(\frac{m\alpha}{\gamma\Delta}\left(\log\left(\frac{\alpha}{\Delta}\right)\log\left(\frac{1}{\gamma}\right)\log\left(\frac{\log(\alpha/\Delta)}{\vartheta}\right)+\log\left(\frac{1}{\epsilon}\right)\right)\right)$.*

It may be sometimes desirable to ignore whether the procedure is successful or not. In this case we will see the output as a mixed state whose density matrix is

$$\rho = \mathrm{Prob}(\mathcal{E}^c)\,|\widetilde{\psi_0}\rangle\langle\widetilde{\psi_0}| + \rho',$$

where $|\widetilde{\psi_0}\rangle$ is the approximate ground state with fidelity at least $1-\epsilon$, which is produced conditional on the event $\mathcal{E}^c$, and $\mathrm{Tr}\rho' = \mathrm{Prob}(\mathcal{E})$. Then this mixed state will have a fidelity lower bounded by

$$\langle\psi_0|\rho|\psi_0\rangle \geq \mathrm{Prob}(\mathcal{E}^c)|\langle\widetilde{\psi_0}|\psi_0\rangle|^2 \geq (1-\vartheta)(1-\epsilon)^2.$$

If we want to achieve $\sqrt{1-\xi}$ fidelity for the mixed state, we can simply let $\vartheta = \epsilon = \xi/3$. Thus the number of queries to U_H and U_I are $\widetilde{\mathcal{O}}(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\xi}))$ and $\widetilde{\mathcal{O}}(\frac{1}{\gamma}\log(\frac{\alpha}{\Delta})\log(\frac{1}{\xi}))$ respectively.

2.5 Optimality of the query complexities

In this section we prove for the ground state preparation algorithms outlined in Section 2.3 and Section 2.4 the number of queries to U_H and U_I are essentially optimal. We will also show our ground energy estimation algorithm has an nearly optimal dependence on the precision. We first prove the following complexity lower bounds:

Theorem 10. *Suppose we have a generic Hamiltonian $H = \sum_k \lambda_k |\psi_k\rangle \langle\psi_k| \in \mathbb{C}^{N\times N}$, where $\lambda_k \leq \lambda_{k+1}$, given through its $(\alpha, m, 0)$-block-encoding U_H, and $\alpha = \Theta(1)$. Also suppose we have an initial state $|\phi_0\rangle$ prepared by circuit U_I, as well as the promises* (P1) *and* (P2). *Then the query complexities of preparing the ground state $|\psi_0\rangle$ of H to fidelity at least $\sqrt{3}/2$ satisfy*

1. *When $\Delta = \Omega(1)$, and $\gamma \to 0^+$, the number of queries to U_H is $\Omega(1/\gamma)$;*
2. *When $\gamma = \Omega(1)$, and $\Delta \to 0^+$, the number of queries to U_H is $\Omega(1/\Delta)$;*
3. *When $\Delta = \Omega(1)$, and $\gamma \to 0^+$, it is not possible to accomplish the above task using $\mathcal{O}(1/\gamma^{1-\theta})$ queries to U_I and $\mathcal{O}(\mathrm{poly}(1/\gamma))$ queries to U_H for any $\theta > 0$.*

Proof. We prove all three lower bounds by applying the ground state preparation algorithm to the unstructured search problem. In the unstructured search problem we try to find a n-bit string t marked out by the oracle

$$U_t = I - 2|t\rangle\langle t|.$$

It is proved for this problem the number of queries to U_t to find t with probability $1/2$ is lower bounded by $\Omega(\sqrt{N})$ where $N = 2^n$ [31].

This problem can be seen as a ground state preparation problem. We find that $|t\rangle$ is the ground state of U_t, which is at the same time a unitary and therefore an $(1, 0, 0)$-block-encoding of itself. Therefore U_t serves as the U_H in the theorem. The spectral gap is 2. Also, let

$$|u\rangle = \frac{1}{\sqrt{N}} \sum_s |s\rangle$$

be the uniform superposition of all n-strings, then we have $\langle u|t\rangle = \frac{1}{\sqrt{N}}$, and $|u\rangle$ can be efficiently prepared by the Hadamard transform since $\mathrm{H}^{\otimes n}|0^n\rangle = |u\rangle$. Therefore $\mathrm{H}^{\otimes n}$ serves as the U_I described in the theorem.

If the ground state preparation problem can be solved with $o(1/\gamma)$ queries to U_H for fixed Δ to produce an approximate ground state with fidelity at least $\sqrt{3}/2$, then from the above setup we have $\gamma = 1/\sqrt{N}$, and we can first find the approximate ground state and then measure in the computational basis, obtaining t with probability at least 3/4. Therefore the unstructured search problem can be solved with $o(\sqrt{N})$ queries to the oracle U_t, which is impossible. Thus we have proved the first lower bound in our theorem.

To prove the second lower bound we want to create a situation in which the overlap is bounded from below by a constant but the gap vanishes. We need to introduce the Grover diffusion operator

$$D = I_n - 2\left|u\right\rangle\left\langle u\right|. \tag{2.3}$$

which can be efficiently implemented. Then we define

$$H(\tau) = (1-\tau)D + \tau U_t, \tag{2.4}$$

and consider $H(1/2)$. Because both $\mathrm{span}(\left|u\right\rangle, \left|t\right\rangle)$ and its orthogonal complement are invariant subspaces of D and U_t, and both operators become the identity operator when restricted to the orthogonal complement of $\mathrm{span}(\left|u\right\rangle, \left|t\right\rangle)$, we only need to look for the ground state in the 2-dimensional subspace $\mathrm{span}(\left|u\right\rangle, \left|t\right\rangle)$. In this subspace, relative to the basis $\{\left|u\right\rangle, \left|t\right\rangle\}$, the matrix representation of $H(1/2)$ is

$$\begin{pmatrix} 0 & -\langle u|t\rangle \\ -\langle t|u\rangle & 0 \end{pmatrix} = -\frac{1}{\sqrt{N}}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Therefore the ground state of $H(1/2)$ is

$$\left|\Psi\right\rangle = \frac{\left|u\right\rangle + \left|t\right\rangle}{\sqrt{2+\frac{2}{\sqrt{N}}}}.$$

and therefore $\langle\Psi|u\rangle = \langle\Psi|t\rangle = 1/\sqrt{2} + \mathcal{O}(1/\sqrt{N})$ for large N. Furthermore, the gap is $\Delta(1/2) = 2/\sqrt{N}$.

Therefore $\left|t\right\rangle$ can be prepared in the following way: we first prepare the ground state of $H(1/2)$, whose block-encoding is easy to construct using one application of U_t. The resulting approximate ground state we denote by $|\widetilde{\Psi}\rangle$. Then we measure $|\widetilde{\Psi}\rangle$ in the computational basis. If there is some non-vanishing probability of obtaining t then we can boost the success probability to above 1/2 by repeating the procedure and verifying using U_t.

If the second lower bound in the theorem does not hold, then $|\widetilde{\Psi}\rangle$ can be prepared with $o(1/\Delta(1/2)) = o(\sqrt{N})$ queries to the block-encoding of $H(1/2)$ and therefore the same number of queries to U_t. Because the angle corresponding to fidelity is the great-circle distance on the unit sphere, we have the triangle inequality (using that $|\langle\widetilde{\Psi}|\Psi\rangle| \geq \sqrt{3}/2$)

$$\arccos|\langle\widetilde{\Psi}|t\rangle| \leq \arccos|\langle\Psi|t\rangle| + \arccos|\langle\widetilde{\Psi}|\Psi\rangle| \leq \frac{5\pi}{12} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

Therefore for large N we have $|\langle\widetilde{\Psi}|t\rangle| \geq \cos(5\pi/12) + \mathcal{O}(1/\sqrt{N}) > 1/4$. The probability of getting t when performing measurement is at least $1/16$. Therefore we can boost the success probability to above $1/2$ by $\mathcal{O}(1)$ repetitions and verifications. The total number of queries to U_t is therefore $o(\sqrt{N})$. Again, this is impossible. Therefore we have proved the second lower bound in our theorem.

For the last lower bound we need to create some trade off between the gap and the overlap. We consider preparing the ground state of the Hamiltonian $H(1/2 - N^{-1/2+\delta})$, $0 < \delta < 1/6$, whose block-encoding can be efficiently constructed with a single application of U_t, as an intermediate step. It is shown in Appendix 2.9 that the ground state is

$$|\Phi\rangle = |u\rangle + \frac{1}{4}N^{-\delta}|t\rangle + \mathcal{O}(N^{-2\delta}). \tag{2.5}$$

Therefore

$$\gamma_u = |\langle\Phi|u\rangle| = 1 + \mathcal{O}(N^{-2\delta}), \quad \gamma_t = |\langle\Phi|t\rangle| = \frac{1}{4}N^{-\delta} + \mathcal{O}(N^{-2\delta}).$$

Also we show in Appendix 2.9 that the gap is

$$\Delta(1/2 - N^{-1/2+\delta}) = 4N^{\delta-1/2} + \mathcal{O}(N^{-1/2-\delta}). \tag{2.6}$$

We first apply the algorithm described in Section 2.3 to prepare the ground state of $H(1/2 - N^{-1/2+\delta})$ to fidelity $1 - N^{-2\delta}/128$. Using the overlap γ_u and the gap in (2.6), the approximate ground state, denoted by $|\widetilde{\Phi}\rangle$, can be prepared with $\mathcal{O}(N^{1/2-\delta}\log(N))$ queries to the block-encoding of $H(1/2 - N^{-1/2+\delta})$, and therefore the same number of queries to U_t.

The overlap between $|\widetilde{\Phi}\rangle$ and $|t\rangle$ can again be bounded using the triangle inequality

$$\begin{aligned}\arccos|\langle\widetilde{\Phi}|t\rangle| &\leq \arccos|\langle\Phi|t\rangle| + \arccos|\langle\widetilde{\Phi}|\Phi\rangle| \\ &\leq \arccos\left(\frac{N^{-\delta}}{4}\right) + \arccos\left(1 - \frac{N^{-2\delta}}{128}\right) + \mathcal{O}(N^{-2\delta}) \\ &\leq \frac{\pi}{2} - \frac{N^{-\delta}}{4} + \sqrt{2 \times \frac{N^{-2\delta}}{128}} + \mathcal{O}(N^{-2\delta}) \\ &= \frac{\pi}{2} - \frac{N^{-\delta}}{8} + \mathcal{O}(N^{-2\delta}).\end{aligned}$$

Therefore we have

$$\widetilde{\gamma}_t = |\langle\widetilde{\Phi}|t\rangle| \geq \frac{N^{-\delta}}{8} + \mathcal{O}(N^{-2\delta}).$$

If the last lower bound in our theorem does not hold, we can then prepare the ground state of U_t by using the initial state $|\widetilde{\Phi}\rangle$ only $\mathcal{O}(1/\widetilde{\gamma}_t^{1-\theta})$ times for some $\theta > 0$, and the number of queries to U_t at this step, i.e. not including the queries used for preparing $|\widetilde{\Phi}\rangle$, is $\mathcal{O}(1/\widetilde{\gamma}_t^{p})$ for some $p > 0$. Therefore the total number of queries to U_t is

$$\mathcal{O}\left(\frac{N^{1/2-\delta}\log(N)}{\widetilde{\gamma}_t^{1-\theta}} + \frac{1}{\widetilde{\gamma}_t^{p}}\right) = \mathcal{O}(N^{1/2-\delta\theta}\log(N) + N^{\delta p}).$$

This complexity must be $\Omega(N^{1/2})$ according to the lower bound for unstructured search problem. Therefore we need $\delta p \geq 1/2$. However we can choose δ to be arbitrarily small, and no finite p can satisfy this condition. Hence we have a contradiction. This proves the last lower bound in our theorem. □

When we look at the query complexities of the ground state preparation algorithms in Secs. 2.3 and 2.4, we can use $\widetilde{\mathcal{O}}$ notation to hide the logarithmic factors, and both algorithms use $\widetilde{\mathcal{O}}(\frac{\alpha}{\gamma\Delta})$ queries to U_H and $\widetilde{\mathcal{O}}(\frac{1}{\gamma})$ queries to U_I when we want to achieve some fixed fidelity. Given the lower bound in Theorem 10 we can see the algorithm with a priori bound for ground energy essentially achieves the optimal dependence on γ and Δ. The algorithm without a priori bound for ground energy achieves the same complexity modulo logarithmic factors, while using less information. This fact guarantees that the dependence is also nearly optimal.

We will then prove the nearly optimal dependence of our ground energy estimation algorithm on the precision h. We have the following theorem:

Theorem 11. *Suppose we have a generic Hamiltonian $H = \sum_k \lambda_k |\psi_k\rangle\langle\psi_k| \in \mathbb{C}^{N\times N}$, where $\lambda_k \leq \lambda_{k+1}$, given through its $(\alpha, m, 0)$-block-encoding U_H, and $\alpha = \Theta(1)$. Also suppose we have an initial state $|\phi_0\rangle$ prepared by circuit U_I, as well as the promise that $|\langle\phi_0|\psi_0\rangle| = \Omega(1)$. Then estimating the ground energy to precision h requires $\Omega(1/h)$ queries to U_H.*

This time we convert the quantum approximate counting problem, which is closely related to the unstructured search problem, into an eigenvalue problem. The quantum approximate counting problem is defined in the following way. We are given a set of n-bit strings $S \subset \{0,1\}^n$ specified by the oracle U_f satisfying

$$U_f|x\rangle = \begin{cases} -|x\rangle & x \in S, \\ |x\rangle & x \notin S, \end{cases}$$

for any $x \in \{0,1\}^n$. We want to estimate the size $|S|/N$ up to relative error ϵ. It has been proven that this requires $\Omega\left(\frac{1}{\epsilon}\sqrt{\frac{N}{|S|}}\right)$ queries to U_f for $|S| = o(N)$ [125, Theorem 1.13], where $N = 2^n$, for the success probability to be greater than 3/4, and this lower bound can be achieved using amplitude estimation [42].

We convert this problem into an eigenvalue problem of a block-encoded Hamiltonian. Let $|u\rangle$ be the uniform superposition of the computational basis and D be the Grover diffusion operator defined in (2.3). Then define the following $(n+1)$-qubit unitary (H is the Hadamard gate)

$$U_H = (\mathrm{H} \otimes I_n)[|0\rangle\langle 0| \otimes D - |1\rangle\langle 1| \otimes (U_f D U_f)](\mathrm{H} \otimes I_n),$$

which can be implemented using two applications of controlled-U_f. We define

$$H = (\langle 0| \otimes I_n) U_H (|0\rangle \otimes I_n) = \frac{1}{2}(D - U_f D U_f).$$

Note that here H is given in its $(1,1,0)$-block-encoding U_H. Let

$$|u\rangle = a\,|u_0\rangle + \sqrt{1-a^2}\,|u_1\rangle$$

where the unit vectors $|u_0\rangle$ and $|u_1\rangle$ satisfy

$$U_f\,|u_0\rangle = -\,|u_0\rangle\,, \quad U_f\,|u_1\rangle = |u_1\rangle\,,$$

then we find $a = \sqrt{|S|/N}$. We only need to estimate the value of a to precision $\mathcal{O}(\epsilon' \sqrt{N/|S|})$ in order to estimate $|S|/N$ to precision ϵ'.

We analyze the eigenvalues and eigenvectors of H. It can be verified that $\{|u_0\rangle, |u_1\rangle\}$ span an invariant subspace of H, and relative to this orthonormal basis H is represented by the matrix

$$\begin{pmatrix} 0 & -2a\sqrt{1-a^2} \\ -2a\sqrt{1-a^2} & 0 \end{pmatrix}.$$

In the orthogonal complement of this subspace, H is simply the zero matrix. Therefore H has only two non-zero eigenvalues $\pm 2a\sqrt{1-a^2}$ corresponding to eigenvectors

$$|\psi_\mp\rangle = \frac{1}{\sqrt{2}}(|u_0\rangle \mp |u_1\rangle).$$

The ground state of H is therefore $|\psi_+\rangle$ with ground energy $-2a\sqrt{1-a^2}$. We can use $|u\rangle$ as the initial state, with an overlap $\langle \psi_+ | u\rangle = \frac{1}{\sqrt{2}}(a + \sqrt{1-a^2}) \geq \frac{1}{\sqrt{2}}$.

We use this Hamiltonian to prove Theorem 11:

Proof. Assume toward contradiction that there exists an algorithm that estimates the ground energy to precision h using only $o(1/h)$ queries to U_H. Then we use this algorithm to estimate the ground energy of the block-encoded Hamiltonian constructed above, for $a = o(1)$, which means $|S| = o(N)$. Estimating $2a\sqrt{1-a^2}$ to precision $\mathcal{O}(h)$ enables us to estimate a to precision $\mathcal{O}(h)$. Setting $h = \epsilon'\sqrt{N/|S|}$, then this algorithm can estimate $|S|/N$ to precision ϵ', with success probability at least 3/4. Since we are interested in the relative error we set $\epsilon' = \epsilon|S|/N$. Therefore the whole procedure uses only $o(1/h) = o(\frac{1}{\epsilon}\sqrt{\frac{N}{|S|}})$ queries to U_H and therefore twice the amount of queries to U_f. This contradicts the lower bound for the approximate counting problem in [125]. □

Remark 12. *Theorem 11 can also be viewed as a consequence of the optimality of the quantum phase estimation algorithm [37]. If instead of the block-encoding U_H we have $e^{-i\tau H}$ as the oracle for some τ such that $|\tau|\|H\| \leq \pi$, then even when given the exact ground state of H, [37, Lemma 3] gives a query complexity lower bound $\Omega(1/h)$ for estimating the ground energy to within additive error h. This provides a different proof of the above theorem, since $e^{-i\tau H}$ and the block-encoding of H are interconvertible: one can efficiently implement $e^{-i\tau H}$ via Hamiltonian simulation starting from a block-encoding of H [114], and can efficiently obtain a block-encoding of H by querying $e^{-i\tau H}$ according to [85, Corollary 71].*

2.6 Low-energy state preparation

It is known that estimating the spectral gap Δ is a difficult task [10, 66, 30]. Our algorithm for finding ground energy, as discussed in Theorem 8, does not depend on knowing the spectral gap. However both of our algorithms for preparing the ground state in Theorem 6 and Corollary 9 require a lower bound of the spectral gap. We would like to point out that if we only want to produce a low-energy state $|\psi\rangle$, making $\langle\psi|H|\psi\rangle \leq \mu$ for some $\mu > \lambda_0$, as in [134], then this can be done without any knowledge of the spectral gap. In fact this is even possible for when the ground state is degenerate.

To do this, we need to first assume we have a normalized initial state $|\phi_0\rangle$ with non-trivial overlap with the low-energy eigen-subspaces. Quantitatively this means for some $\gamma, \delta > 0$, if we expand the initial state in the eigenbasis of H, obtaining $|\phi_0\rangle = \sum_k \alpha_k |\psi_k\rangle$, then $\sum_{k:\lambda_k \leq \mu-3\delta} |\alpha_k|^2 \geq \gamma^2$. Then we can use the block-encoded projection operator in (2.2) to get

$$|\psi'\rangle = (\langle 0^{m+3}| \otimes I)\mathrm{PROJ}(\mu - 2\delta, \delta, \epsilon')(|0^{m+3}\rangle \otimes |\phi_0\rangle),$$

for some precision ϵ'. Now we expand $|\psi'\rangle$ in the eigenbasis to get $|\psi'\rangle = \sum_k \beta_k |\psi_k\rangle$, and denote $|\varphi'\rangle = \sum_{k:\lambda_k<\mu-\delta} \beta_k |\psi_k\rangle$. We then have, because of the approximation to the sign function,

$$\| |\psi'\rangle - |\varphi'\rangle \| \leq \frac{\epsilon'}{2}, \quad \langle\varphi'|\varphi'\rangle \geq \gamma^2(1-\frac{\epsilon'}{2})^2, \quad \langle\varphi'|H|\varphi'\rangle \leq (\mu-\delta)\langle\varphi'|\varphi'\rangle .$$

From the above bounds we further get

$$\frac{\langle\psi'|H|\psi'\rangle}{\langle\psi'|\psi'\rangle} \leq \frac{\langle\varphi'|H|\varphi'\rangle + \|H\|\epsilon' + \|H\|\epsilon'^2/4}{\langle\varphi'|\varphi'\rangle - \epsilon'} \leq \frac{\mu - \delta + \frac{\alpha\epsilon'+\alpha\epsilon'^2/4}{\gamma^2(1-\epsilon'/2)^2}}{1-\frac{\epsilon'}{\gamma^2(1-\epsilon'/2)^2}}.$$

Now denoting $|\psi\rangle = |\psi'\rangle / \| |\psi'\rangle \|$ we can make $\langle\psi|H|\psi\rangle \leq \mu$ by choosing $\epsilon' = \mathcal{O}(\gamma^2\delta/\alpha)$. Therefore the total number of queries to U_H required is $\mathcal{O}(\frac{1}{\delta\gamma}\log(\frac{\alpha}{\delta\gamma}))$ and the number of queries to U_I is $\mathcal{O}(\frac{1}{\gamma})$.

From this we can see that if the initial state $|\phi_0\rangle$ has a overlap with the the ground state that is at least γ, and we want to prepare a state with energy upper bounded by $\lambda_0 + \delta$, the required number of queries to U_H and U_I are $\mathcal{O}(\frac{1}{\delta\gamma}\log(\frac{\alpha}{\delta\gamma}))$ and $\mathcal{O}(\frac{1}{\gamma})$ respectively. If we do not know the ground energy beforehand we can use the algorithm in Theorem 8 to estimate it first. Note that none of these procedures assumes a spectral gap.

2.7 Discussions

In this work we proposed an algorithm to prepare the ground state of a given Hamiltonian when a ground energy upper bound is known (Theorem 6), an algorithm to estimate the ground energy based on binary search (Theorem 8), and combining these two to get an algorithm to prepare the ground state without knowing an upper bound a priori (Corollary 9). By solving the unstructured search problem and the approximate counting problem through preparing the ground state, we proved that the query complexities for the tasks above cannot be substantially improved, as otherwise the complexity lower bound for the two problems would be violated.

All our algorithms are based on the availability of the block-encoding of the target Hamiltonian. This is a non-trivial task but we know it can be done for many important settings. For example, Childs et al. proposed an LCU approach to block-encode the Hamiltonian of a quantum spin system [58], in which the Hamiltonian is decomposed into a sum of Pauli matrices. In [116], Low and Wiebe outlined the methods to construct block-encoding of Hubbard Hamiltonian with long-range interaction, and of quantum chemistry Hamiltonian in plane-wave basis, both using fast-fermionic Fourier transform (FFFT) [28]. The FFFT can be replaced by a series of Givens rotations which gives lower circuit depth and better utilizes limited connectivity [104, 98]. Any sparse Hamiltonian whose entries can be efficiently computed can also be block-encoded using a quantum walk operator [33, 34, 59].

We remark that the quantum circuit used in our method for ground energy estimation can be further simplified. The main obstacle to applying this method to near-term devices is the need of amplitude estimation, which requires phase estimation. It is possible to replace amplitude estimation by estimating the success probability classically. In the context of binary amplitude estimation in Lemma 7, we need to determine whether the success amplitude is greater than $3\gamma/4$ or smaller than $\gamma/4$. This can be turned into a classical hypothesis testing to determine whether the success probability is greater than $9\gamma^2/16$ or smaller than $\gamma^2/16$. A simple Chernoff bound argument tells us that we need $\mathcal{O}(\log(1/\vartheta)/\gamma^2)$ samples to distinguish the two cases with success probability at least $1-\vartheta$, as opposed to the $\mathcal{O}(\log(1/\vartheta)/\gamma)$ complexity in amplitude estimation.

In this approach, the only quantum circuit we need to use is the one in (2.2). The circuit depth is therefore only $\mathcal{O}((\alpha/h)\log(1/\gamma))$. It also does not require the $\mathcal{O}(\log(1/\gamma))$ qubits that are introduced as a result of using amplitude estimation. These features make it suitable for near-to-intermediate term devices.

In [112] we proposed an eigenstate filtering method (similar in spirit to the method proposed in Section 2.3), and we combined it with quantum Zeno effect [57, 39] to solve the quantum linear system problem. The resulting algorithm utilizes the fact that the desired eigenstate along the eigenpath always corresponds to the eigenvalue 0. In the setting of quantum Zeno effect based state preparation, in which we have a series of Hamiltonians and wish to incrementally prepare the ground state of each of them, our algorithm in Theorem 6 can be used to go from the ground state of one Hamiltonian to the next one, provided that we have a known upper bound for the ground energy. In the absence of such an upper bound, there is the possibility of using the algorithm in Corollary 9 to solve this problem. However in this setting we only want to use the initial state once for every Hamiltonian, since preparing the initial state involves going through the ground state of all previous Hamiltonians. This presents a challenge and is a topic for our future work.

It is worth pointing out that none of the Hamiltonians used in the proofs of lower bounds in Section 2.5 is a local Hamiltonian, and therefore our lower bounds do not rule out the possibility that if special properties such as locality are properly taken into consideration, better complexities can be achieved.

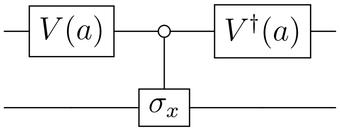
2.8 An example of block-encoding and constructing the reflector

In this section we use σ_x, σ_y, and σ_z to denote the three Pauli matrices. We use H to denote the Hadamard gate. We consider a single-qubit illustrative example of block-encoded matrix and obtain the corresponding reflector through QSP.

The matrix we consider is

$$H(a) = a\sigma_x + (1-a)I,$$

for $0 \leq a \leq 1$. Its block-encoding can be using the following circuit



where

$$V(a) = \begin{pmatrix} \sqrt{a} & -\sqrt{1-a} \\ \sqrt{1-a} & \sqrt{a} \end{pmatrix}.$$

We denote the above circuit by $U_H(a)$. This is a $(\alpha, m, 0)$-block-encoding of $H(a)$ where $\alpha = 1$ and $m = 1$, since we can readily check that

$$(\langle 0| \otimes I)U_H(a)(|0\rangle \otimes I) = H(a).$$

The eigendecomposition of $H(a)$ is

$$H(a) = |+\rangle \langle +| + (1-2a) |-\rangle \langle -| \,,$$

with eigenvalues $\lambda_+(a) = 1, \lambda_-(a) = 1-2a$. Our goal is to implement the reflector

$$R_{<0}(a) = -\mathrm{sign}(H(a)) = -|+\rangle \langle +| - \mathrm{sign}(1-2a) |-\rangle \langle -| \,.$$

To do this we need an odd polynomial $S(x;\delta,\epsilon)$ introduced in Lemma 3. Instead of the construction done in Ref. [115] we use the Remez algorithm [138] to obtain this polynomial. We choose $\delta = 0.2$ and the L^∞ error of the residual is required to be less than 10^{-4}, i.e. $\epsilon \leq 10^{-4}$.

Given the polynomial $S(x;\delta,\epsilon)$, using the optimization method proposed in Ref. [70], we find a polynomial $P(x) \in \mathbb{C}[x]$ of odd degree d such that

$$\max_{x\in[-1,1]} |\operatorname{Re} P(x) - S(x;\delta,\epsilon)| \leq \epsilon',$$

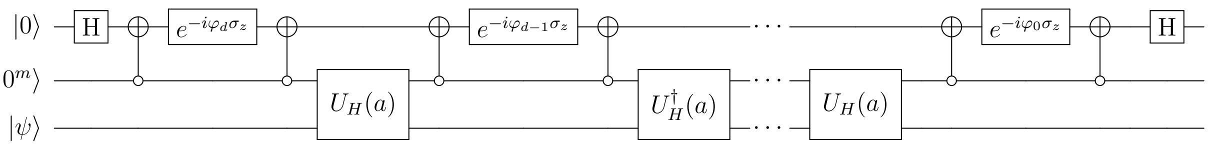


Figure 2.1: The circuit implementing the polynomial eigenvalue transformation through QSP for an odd polynomial with phase factors $\{\varphi_j\}_{j=0}^d$. H is the Hadamard gate and σ_z is the Pauli-Z gate.

where $P(x)$ is characterized by a sequence of phase factors $\{\varphi_j\}_{j=0}^d$ satisfying

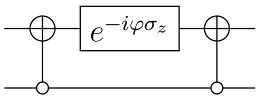
$$\begin{pmatrix} P(x) & \cdot \\ \cdot & \cdot \end{pmatrix} = e^{i\varphi_0\sigma_z}\prod_{j=1}^{d}[R(x)e^{i\varphi_j\sigma_z}], \tag{2.7}$$

where

$$R(x) = \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix}.$$

The existence of the phase factors is guaranteed by [86, Theorem 5]. Ref. [70] uses quasi-Newton method to solve a least squares problem to obtain these phase factors, and we terminate the iteration only when L^∞ error of the residual of the real part is smaller than $\epsilon' = 10^{-4}$.

The circuit in Figure 2.1 with phase factors $\{\varphi_j\}_{j=0}^d$ implements the transformation $H/\alpha \mapsto \operatorname{Re} P(H/\alpha) \approx S(H/\alpha;\delta,\epsilon)$. The various components of this circuit are explained in detail in [86, Figure 3]. An important component of this circuit is



where the first register has one qubit, the second register has m-qubits, and the open bullet indicates control-on-zero for multiple control qubits. This component implements the operator

$$|0\rangle\langle 0| \otimes (e^{i\varphi(2|0^m\rangle\langle 0^m|-I)}) + |1\rangle\langle 1| \otimes (e^{-i\varphi(2|0^m\rangle\langle 0^m|-I)}).$$

For a detailed discussion see [86, Corollary 11].

Using the above circuit, Lemma 5 guarantees that when the eigenvalues of $H(a)$ are contained in $[-1,-\delta] \cup [\delta, 1]$, we will have a good approximation of $R_{<0}(a)$.

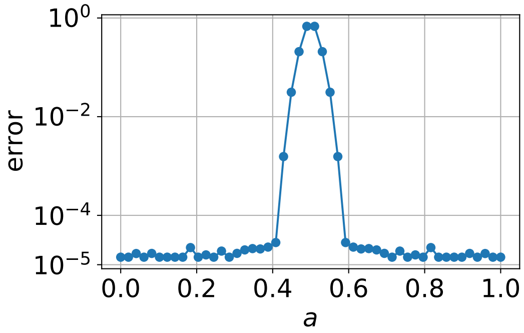


Figure 2.2: The error of implementing $R_{<0}(a)$ for $a \in [0,1]$ using QSP, with polynomial $S(x;\delta,\epsilon)$ where $\delta = 0.2$ and ϵ is of the order of 10^{-4}. The vertical axis uses logarithmic scale.

However when at least one eigenvalue, which in our case can only be $\lambda_-(a) = 1-2a$, is in $(-\delta,\delta)$, or in other words when $a \in (0.4, 0.6)$, there is no such guarantee. We plot the operator norm error between the approximate reflector obtained through QSP and the exact reflector $R_{<0}(a)$ in Figure 2.2. It can be seen in the figure that the error is smaller than 10^{-4} everywhere except for $a \in (0.4, 0.6)$, where the error spikes.

2.9 Gap and overlap in the unstructured search problem

In this appendix we compute the spectral gap of the Hamiltonian $H(1/2 - N^{-1/2+\delta})$ for $H(\tau)$ defined in (2.4), $0 < \delta < 1/6$, and the overlap between its ground state and $|u\rangle$ and $|t\rangle$ defined in Section 2.5.

The first thing we should realize is that we only need to care about the subspace of the Hilbert space spanned by $|u\rangle$ and $|t\rangle$. In the orthogonal complement of this subspace $H(\tau)$ is simple a multiple of identity. In this subspace, with respect to the non-orthogonal basis $\{|u\rangle, |t\rangle\}$, the operator $H(1/2 - N^{-1/2+\delta})$ is represented by the following matrix

$$N^{\delta-1/2}\begin{pmatrix} -2 & -(N^{-\delta}+2N^{-1/2}) \\ -(N^{-\delta}-2N^{-1/2}) & 2 \end{pmatrix}. \tag{2.8}$$

Direct calculation shows the eigenvalues are

$$\lambda_{\pm} = \pm N^{\delta-1/2}\sqrt{4+N^{-2\delta}-4N^{-1}} = \pm N^{\delta-1/2}(2+\frac{1}{4}N^{-2\delta}+\mathcal{O}(N^{-4\delta})).$$

Thus we obtain the spectral gap in (2.6). To simplify notation we let $\widetilde{\lambda} = N^{1/2-\delta}\lambda_+$. We then compute the ground state. We first find an eigenvector corresponding to λ_-

$$\begin{aligned}
|\chi\rangle &= N^{\delta}((N^{-\delta}+2N^{-1/2})\,|u\rangle + (-2+\widetilde{\lambda})\,|t\rangle) \\
&= (1+2N^{\delta-1/2})\,|u\rangle + (\frac{1}{4}N^{-\delta}+\mathcal{O}(N^{-3\delta}))\,|t\rangle \\
&= |u\rangle + \frac{1}{4}N^{-\delta}\,|t\rangle + \mathcal{O}(N^{\delta-1/2}).
\end{aligned}$$

We still need to normalize $|\chi\rangle$. The normalization factor is

$$\begin{aligned}
\|\,|\chi\rangle\,\| &= \sqrt{(1+2N^{\delta-1/2})^2+(\frac{1}{4}N^{-\delta}+\mathcal{O}(N^{-3\delta}))^2+\frac{2}{\sqrt{N}}(1+2N^{\delta-1/2})(\frac{1}{4}N^{-\delta}+\mathcal{O}(N^{-3\delta}))} \\
&= 1+\mathcal{O}(N^{-2\delta}).
\end{aligned}$$

Note that the third term under the square root comes from the overlap between $|u\rangle$ and $|t\rangle$, and it does not play an important role asymptotically. Therefore normalizing we have the expression for the normalized eigenstate (2.5).

Chapter 3

Quantum algorithms for the early fault-tolerant setting

Under suitable assumptions, the quantum phase estimation (QPE) algorithm is able to achieve Heisenberg-limited precision scaling in estimating the ground state energy. However, QPE requires a large number of ancilla qubits and large circuit depth, as well as the ability to perform inverse quantum Fourier transform, making it expensive to implement on an early fault-tolerant quantum computer. We propose an alternative method to estimate the ground state energy of a Hamiltonian with Heisenberg-limited precision scaling, which employs a simple quantum circuit with one ancilla qubit, and a classical post-processing procedure. Besides the ground state energy, our algorithm also produces an approximate cumulative distribution function of the spectral measure, which can be used to compute other spectral properties of the Hamiltonian.

3.1 Introduction

Estimating the ground state energy of a quantum Hamiltonian is of immense importance in condensed matter physics, quantum chemistry, and quantum information. The problem can be described as follows: we have a Hamiltonian H, acting on n qubits, with the eigendecomposition

$$H = \sum_{k=0}^{K-1} \lambda_k \Pi_k,$$

where Π_k is the projection operator into the λ_k-eigensubspace, and λ_k's are increasingly ordered. Each eigenvalue may be degenerate, i.e. the rank of Π_k can be more than one. We assume we can access the Hamiltonian H through the time evolution operator $e^{-i\tau H}$ for some fixed τ. Our goal is to estimate the ground state energy λ_0 to within additive error ϵ.

Some assumptions are needed as otherwise this problem is QMA-hard [102, 100, 130, 6]. We assume we are given a state described by its density matrix ρ. Let $p_k = \mathrm{Tr}[\rho\Pi_k]$. Then if p_0 (i.e. the overlap between the initial state and the ground state) is reasonably large we can solve the ground state energy estimation problem efficiently. This assumption is reasonable in many practical settings. For example, in quantum chemistry, the Hartree-Fock method usually yields an approximate ground state that is easy to prepare on a quantum computer. At least for relatively small molecular systems, the Hartree-Fock state can often have a large overlap with the exact ground state [165]. Therefore we may use the Hartree-Fock solution as ρ in this setting. Other candidates of ρ that can be relatively easily prepared on quantum computers have been discussed in Refs. [25, 155, 165], and an overview of methods to choose ρ can be found in [119, Section V.A.2].

The computational complexity of this task depends on the desired precision ϵ. Even in the ideal case where the exact ground state is given, this dependence cannot be better than linear in ϵ^{-1} for generic Hamiltonians [24]. This limit is called the Heisenberg limit [88, 87, 182, 181] in quantum metrology. This notion is closely related to the time energy uncertainty principle [7, 8, 54, 24]. This optimal scaling can be achieved using the quantum phase estimation (QPE) algorithm [101], which we will discuss in detail later.

Much work has been done to develop the algorithms for ground state energy estimation both for near-term quantum devices [133, 120, 129, 96], and fully fault-tolerant quantum computers [3, 134, 81, 111]. Relatively little work has been done for early fault-tolerant quantum computers [47, 27, 41, 108] , which we expect to be able to accomplish much more complicated tasks than current and near-term devices, but still place significant limitations on the suitable algorithms. Refs. [103, 47] carried out careful resource cost estimation of performing QPE for the Hubbard model using surface code to perform quantum error correction. These are to our best knowledge the only works that addressed ground state energy estimation in the context of early fault-tolerant quantum computers.

To be specific, we expect such early fault-tolerant quantum computers to have the following characteristics: (1) The number of logical qubits are limited. (2) It is undesirable to have a large number of controlled operations. (3) It is a priority to reduce the circuit depth, e.g. it is better to run a circuit of depth $\mathcal{O}(D)$ for $\mathcal{O}(M)$ times than to run a circuit of depth $\mathcal{O}(DM)$ for a constant number of times, even if using the shorter circuit entails some additional poly-logarithmic factors in the total runtime.

In this context, the textbook version of QPE (see e.g. Refs. [62, 126]), which uses multiple ancilla qubits to store the phase and relies on inverse quantum Fourier transform (QFT), has features that are not desirable on early fault-tolerant quantum computers. Some variants of QPE have been developed to achieve high confidence level [105, 135, 124], which can be important in many applications. However, such modifications require even more ancilla qubits to store multiple estimates of the phase and an additional coherent circuit to take perform logical operations. Another possible way to achieve high confidence level is to utilize a resource state ([26, Section II B]) to implement a Kaiser window filter [145]. This approach requires the same number of ancilla qubits as the textbook version of QPE.

Due to the above considerations, we focus on the variants of QPE that use only very few ancilla qubits (in fact, all algorithms below use only one ancilla qubit). Kitaev's algorithm (see e.g. [102]) uses a simple quantum circuit with one control qubit to determine each bit of the phase individually. However this method, together with many other algorithms based on it [167, 170], are designed for phase estimation with an eigenstate given exactly, which is different from our goal. The semi-classical Fourier transform [89] can simulate QFT+measurement (meaning all qubits are measured in the end) with only one-qubit gates, classical control and post-processing, thus trading the expensive quantum resource for inexpensive classical operations. One can replace the inverse QFT with the semi-classical Fourier transform, and this results in a phase estimation algorithm that uses only one ancilla qubit [95, 32]. This approach can be seen as a simulation of the multiple-ancilla qubit version of QPE, and is therefore applicable to the case when ρ is not exactly the ground state. Because of these attractive features this is the version of QPE used in Refs. [103, 47]. However, as we will explain below in section 3.1, this type of QPE requires running coherent time evolution for time $\mathcal{O}(p_0^{-1}\epsilon^{-1})$. This leads to large circuit depth when p_0 is small. Moreover, this approach cannot be used together with the resource state discussed earlier because the resource state is not a product state.

In this work, the complexity is measured by the time for which we need to perform time evolution with the target Hamiltonian H. We will use two metrics: (1) the *maximal evolution time*, which is the maximum length of time for which we need to perform (controlled) coherent time evolution, and (2) the *total evolution time*, which is the sum of all the lengths of time we need to perform (controlled) coherent time evolution. They describe respectively the circuit depth and the total runtime. Moreover, we will be primarily concerned with how they depend on the initial overlap p_0 and the precision ϵ. The dependence on the system size n mainly comes indirectly through p_0 and the conversion between the total evolution time and runtime, which we will discuss in more detail later. We present an algorithm that achieves the following goals:

(1) Achieves Heisenberg-limited precision scaling, i.e. the total time for which we run time evolution is $\widetilde{\mathcal{O}}(\epsilon^{-1}\mathrm{poly}(p_0^{-1}))$;

(2) Uses at most one ancilla qubit;

(3) The maximal evolution time is at most $\mathcal{O}(\epsilon^{-1}\mathrm{polylog}(\epsilon^{-1}p_0^{-1}))$.

To our best knowledge our algorithm is the first to satisfy all three requirements. In our algorithm, we sample from a simple quantum circuit, and use the samples to approximately reconstruct the cumulative distribution function (CDF) of the spectral measure associated with the Hamiltonian. We then use classical post-processing to estimate the ground state energy with high confidence. Besides the ground state energy, our algorithm also produces the approximate CDF, which may be of independent interest. In the discussion above we assumed the controlled time evolution can be efficiently done. If controlled time evolution is costly to implement, then based on ideas in Refs. [96, 142, 118, 127], we offer an alternative circuit in Appendix 3.11 which uses two ancilla qubits, with some additional assumptions.

The problem of ground state energy estimation is closely related to that of ground state preparation, but there are important differences. First, having access to a good initial state ρ (with large overlap with the ground state) does not make the energy estimation a trivial task, as even if we have access to the exact ground state the quantum resources required to perform phase estimation can still be significant. Second, ground state energy estimation algorithms do not necessarily involve ground state preparation. This is true for the algorithm in this work as well as in Refs. [81, 111]. Consequently, even though the ground state preparation algorithms generally have a runtime that depends on the spectral gap between the two lowest eigenvalues of the Hamiltonian, the cost of ground state energy estimation algorithms may not necessarily depend on the spectral gap.

We remark that although we characterize the scaling as depending on the overlap p_0, in practice we need to know a lower bound of p_0, which we denote by η. The dependence on p_0 should more accurately be replaced by a dependence on η. To our best knowledge, in order to obtain rigorous guarantee of the performance, the knowledge of η (and that η is not too small) is needed in all previous algorithms related to QPE. This is because in QPE we need the knowledge of η to obtain a stopping criterion. We will briefly explain this using a simple example. Suppose we have a Hamiltonian H on n qubits with eigenvalues λ_k (arranged in ascending order), and eigenstates $|\psi_k\rangle$, and $|\phi_0\rangle$ is an initial guess for the ground state. Furthermore we assume $p_0 = |\langle\phi_0|\psi_0\rangle|^2 = 0.01,\ p_1 = |\langle\phi_0|\psi_1\rangle|^2 = 0.5$. We may idealize QPE as exact energy measurement to simplify discussion. If we have no a priori knowledge of p_0, then performing QPE on the state $|\phi_0\rangle$ will give us λ_1 with probability 1/2. If we repeat this $\lesssim 100$ times most likely all energies we get will be $\geq \lambda_1$. Only when we measure $\gtrsim 100$ times can we reach the correct ground state energy λ_0. Hence if we do not know about a lower bound of p_0, we can never know whether we have stopped the algorithm prematurely.

The main idea of our algorithm is to use a binary search procedure to gradually narrow down the interval in which the ground state energy is located. The key component is a subroutine CERTIFY (Algorithm 3) that distinguishes whether the ground state energy is approximately to the left or right of some given value. This, however, can only be perform up to certain precision, and can fail with non-zero probability. Therefore our search algorithm needs to account for this fuzzy outcome to produce a final result that is correct with probability arbitrarily close to 1. In the CERTIFY procedure, we use a stochastic method to evaluate the cumulative distribution function associated with the spectral density, and this is the key to achieving the Heisenberg scaling. This stochastic method is described in detail in Section 3.3.

Related works

We first briefly analyze the cost of the textbook version of QPE using multiple ancilla qubits. Although this method has features that are not desirable on early fault-tolerant quantum computers, this analysis will nevertheless be helpful for understanding the cost of other variants of QPE. For simplicity we assume $\rho = |\phi\rangle \langle\phi|$ is a pure state, and the ground state $|\psi_0\rangle$ is non-degenerate. Approximately, the QPE performs a projective measurement in the eigenbasis of H. With probability p_0, $|\phi\rangle$ will collapse to the ground state $|\psi_0\rangle$. If this happens the energy register will then give the ground state energy λ_0 to precision ϵ. Therefore we run phase estimation for a total of $\mathcal{O}(p_0^{-1})$ times, and take the instance with the minimum value in the energy register. With high probability this value will be close to λ_0. Each single run takes time $\mathcal{O}(\epsilon^{-1})$. The total runtime cost is therefore $\mathcal{O}(p_0^{-1}\epsilon^{-1})$. For simplicity here we do not consider the runtime needed to prepare $|\phi\rangle$.

The above analysis, however, is overly optimistic. Since we need to repeat the phase estimation procedure for a total of $\mathcal{O}(p_0^{-1})$ times, for an event that only has $\mathcal{O}(p_0)$ probability of happening in a single run, the probability of this event occurring at least once in the total $\mathcal{O}(p_0^{-1})$ repetitions is now $\mathcal{O}(1)$ (which means we cannot ensure that the error happens with sufficient low probability). In our setting, suppose the maximal evolution time is T, then each time we measure the energy register there is a $\mathcal{O}(T^{-1}\epsilon'^{-1})$ probability that the output will be smaller than $\lambda_0 - \epsilon'$. If we choose $T = \mathcal{O}(\epsilon^{-1})$ as discussed above, and we let $\epsilon' = \epsilon/p_0$, then the probability of the minimum of the $\mathcal{O}(p_0^{-1})$ energy register measurement outputs being smaller than

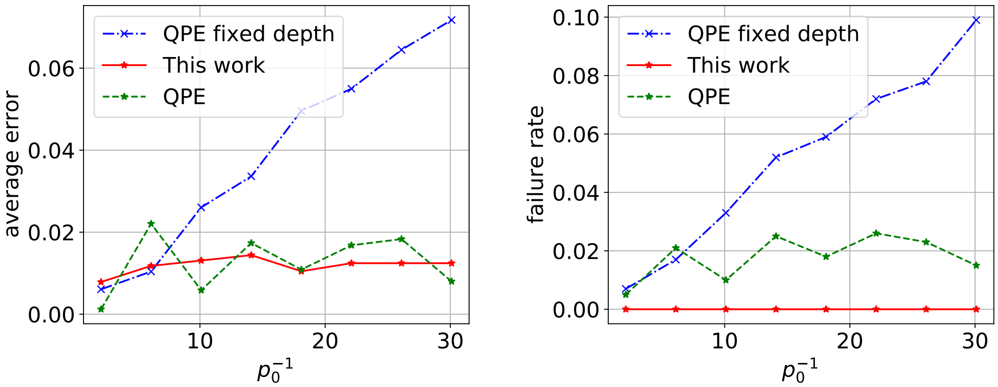


Figure 3.1: Comparing the performance of the textbook version QPE (blue dashed-dotted line) and the method in this work (red solid line) in ground state energy estimation with a fixed maximal evolution time (300 steps of time evolution with H) and decreasing initial overlap p_0. The results are benchmarked against QPE with maximal evolution time proportional to p_0^{-1} (green dashed line). To use QPE, either with fixed or $\mathcal{O}(p_0^{-1})$ maximal evolution time, to estimate the ground state energy, we run QPE for $\mathcal{O}(p_0^{-1})$ times and take the minimum in energy measurement outcomes as the ground state energy estimate. The error is averaged over multiple runs, and the failure rate is the percentage of runs that yield an estimate with error larger than the tolerance 0.04. The Hamiltonian H is the Hubbard Hamiltonian defined in Eq. (3.40) with $U = 10$, and the overlap p_0 is artificially tuned.

$\lambda_0 - \epsilon/p_0$ is only upper bounded by $\mathcal{O}(1)$, and we can no longer control over the probability of the error being larger than ϵ. This means there might be a high probability that the error of the ground state energy in the end will be of order ϵ/p_0 instead of ϵ. For a more formal analysis see [81, Appendix A]. We numerically demonstrate that this is indeed the case in Figure 3.1, in which we show the error increases as p_0 decreases and there is a larger probability of the estimate deviating beyond a prescribed tolerance if the maximal evolution time, or equivalently the circuit depth, for QPE is fixed.

To avoid this, one can instead choose the maximal evolution time to be $T = \mathcal{O}(p_0^{-1}\epsilon^{-1})$. After repeating $\mathcal{O}(p_0^{-1})$ times, the total runtime then becomes $\mathcal{O}(p_0^{-2}\epsilon^{-1})$. The increase in maximal evolution time can prevent the increase of error (see Figure 3.1). However, the extra p_0^{-1} factor increases the circuit depth and is undesirable.

There are several other algorithms based on phase estimation using a single ancilla qubit [167, 170, 128] that are designed for different settings from ours: they assume the availability of an exact eigenstate, or are designed for obtaining the entire spectrum and thus only work for small systems. Ref. [148] proposes a method for estimating the eigenvalues by first estimating $\mathrm{Tr}[\rho e^{-itH}]$ and then performing a classical Fourier transform, but no runtime scaling is provided. The semi-classical Fourier transform [89] simulates the QFT in a classical manner, and the QPE using single ancilla qubit and semi-classical Fourier transform has the same scaling in terms of the maximal evolution time and the total evolution time.

In order to improve the dependence on p_0, we may use the high-confidence versions of the phase estimation algorithm [105, 135, 124]. In this method, the maximal evolution time required can be reduced to $\mathcal{O}(\epsilon^{-1}\log(p_0^{-1}))$, through taking the median of several copies of the energy register in a coherent manner. However, this requires using multiple copies of the energy register, together with an additional quantum circuit to compute the medians coherently that can be difficult to implement. Note that semi-classical Fourier transform can only simulate the measurement outcome and does not preserve coherence, and therefore to our knowledge, the high-confidence version of phase estimation cannot be modified to use only a single qubit. In Ref. [81], the authors used a method called minimum label finding to improve the runtime to $\mathcal{O}(p_0^{-3/2}\epsilon^{-1})$, but the implementation of the minimum label finding with limited quantum resources is again difficult.

Besides these algorithms based on phase estimation, several other algorithms have been developed to solve the ground state energy problem. Ref. [81] proposed a method based on the linear combination of unitaries (LCU) technique that requires running time evolution for duration $\widetilde{\mathcal{O}}(p_0^{-1/2}\epsilon^{-3/2})$ and preparing the initial state $\widetilde{\mathcal{O}}(p_0^{-1/2}\epsilon^{-1/2})$ times.[1] Assuming the Hamiltonian H is available in its block-encoding [114, 49], Ref. [111] uses quantum signal processing [115, 86] with a binary search procedure, which queries the block-encoding $\widetilde{\mathcal{O}}(p_0^{-1/2}\epsilon^{-1})$ times and prepares the initial state $\widetilde{\mathcal{O}}(p_0^{-1/2}\log(\epsilon^{-1}))$ times. To our knowledge, this is the best complexity that has been achieved. However the block-encoding of a quantum Hamiltonian of interest, LCU, and amplitude estimation techniques (used in [111]) are expensive in terms of the number of ancilla qubits, controlled operations, and logical operations needed.

A very different type of algorithms for ground state energy estimation is the variational quantum eigensolver (VQE) [133, 120, 129], which are near-term algorithms and have been demonstrated on real quantum computers. The accuracy of VQE is limited both by the representation power of the variational ansatz, and the capabilities of classical optimization algorithms for the associated non-convex optimization

[1] In this paper we use the following asymptotic notations besides the usual $\mathcal{O}$ notation: we write $f=\Omega(g)$ if $g=\mathcal{O}(f)$; $f=\Theta(g)$ if $f=\mathcal{O}(g)$ and $g=\mathcal{O}(f)$; $f=\widetilde{\mathcal{O}}(g)$ if $f=\mathcal{O}(g\,\mathrm{polylog}(g))$.

problem. Hence unlike aforementioned algorithms, there is no provable performance guarantees for VQE-type methods. In fact some recent results show solving the non-convex optimization problem can be NP-hard [38]. Furthermore, each evaluation of the energy expectation value to precision ϵ requires $\mathcal{O}(\epsilon^{-2})$ samples due to Monte Carlo sampling. This can to some extent be remedied using the methods in [105, 167] at the expense of larger circuit depth requirement.

There are also a few options that can be viewed to be in-between VQE and QPE. The quantum imaginary time evolution (QITE) algorithm [122] uses state tomography turning an imaginary time evolution into a series of real time Hamiltonian evolution problem. Inspired by the classical Krylov subspace method, Refs. [151, 132, 96] propose to solve the ground state energy problem by restricting the Hilbert space to a low dimension space spanned by some eigenstates that are accessible with time evolution. Similar to VQE, no provable complexity upper bound is known for these algorithms, and all algorithms suffer from the ϵ^{-2} scaling due to the Monte Carlo sampling. In fact, the stability of these algorithms remains unclear in the presence of sampling errors.

A more ambitious goal than ground state energy estimation is to estimate the distribution of all eigenvalues weighted by a given initial state ρ [73, 128, 149]. Using a quantum circuit similar to that in Kitaev's algorithm as well as classical post-processing, Ref. [149] proposed an algorithm to solve the quantum eigenvalue estimation problem (QEEP). We henceforth refer to this algorithm as the quantum eigenvalue estimation algorithm (QEEA). Suppose $\|H\| \leq 1/2$, and the interval $[-\pi, \pi]$ is divided into M bins of equal size denoted by $B_j = [-1/2 + j/M, -1/2 + (j+1)/M]$. Then QEEA estimates the quantities $q_j = \sum_{k:\lambda_k \in B_j} p_k$. Although QEEA was not designed for ground state energy estimation, one can use this algorithm to find the leftmost bin in which $q_j \geq p_0/2$, and thereby locate the ground state energy within a bin of size M^{-1}. While the maximal evolution time required scales as $\mathcal{O}(\epsilon^{-1})$, the total evolution time of the original QEEA scales as $\mathcal{O}(\epsilon^{-6})$. We analyze the cost of QEEA in Appendix 3.9, and show that the total runtime can be reduced to $\mathcal{O}(\epsilon^{-4})$ for the ground state energy estimation in a straightforward way, yet this is still costly if high precision is required.

To the extent of our knowledge, none of the existing algorithms achieves all three goals listed on Page 46. Some can have better maximal evolution time or total evolution time requirement, but the advantage always comes at the expense of some other aspects. In Table 3.1 we list the quantum algorithms discussed in this work and whether they satisfy each of the requirements.

In Table 3.2, we compare the maximal evolution time, the number of repetitions (the number of times we need to run the quantum circuit), and the total evolution time needed, using the three qubit-efficient methods that require only one ancilla

Algorithms	Requirements			Other issues
	(1)	(2)	(3)	
QPE (textbook version) [62, 126]	✓	✗	✗	
QPE (high-confidence) [105, 135, 124]	✓	✗	✓	
QPE (semi-classical QFT) [95, 32]	✓	✓	✗	
QPE (iterative) [102]	✓	✓	✓	Needs exact eigenstate ($p_0 = 1$)
The LCU approach [81]	✗	✗	✗	
The binary search approach [111]	✓	✗	✗	
VQE [133, 120, 129]	✗	✓	?	No precision guarantee
QITE [122]	✗	✓	?	Requires state tomography
QEEA [149]	✗	✓	✓	
Krylov subspace methods [151, 132, 96]	✗	✓	?	No precision guarantee
This work	✓	✓	✓	

Table 3.1: Quantum algorithms for estimating the ground state energy and whether they satisfy each of the three requirements on Page 46. We recall that the requirements are (1) achieving the Heisenberg-limited precision scaling, (2) using at most one ancilla qubit, and (3) the maximal evolution time being at most $\mathcal{O}(\epsilon^{-1}\mathrm{polylog}(\epsilon^{-1}p_0^{-1}))$.

qubit.

Finally, in a gate-based setting, the exact relations between the maximal evolution time and the circuit depth, and between the total evolution time and the total runtime, can be affected by the method we use to perform time evolution. Suppose we have access to a unitary circuit that performs $e^{-i\tau H}$ exactly for some fixed τ. Then in order to run coherent time evolution for time T we only need to use a circuit of depth $\mathcal{O}(T)$. Therefore the circuit depth scales linearly with respect to the maximal evolution time. Similarly the total runtime scales linearly with respect to the total evolution time.

	Max evolution time	Repetitions	Total evolution time
This work (Corollary 15)	$\widetilde{\mathcal{O}}(\epsilon^{-1}\mathrm{polylog}(p_0^{-1}))$	$\widetilde{\mathcal{O}}(p_0^{-2}\mathrm{polylog}(\epsilon^{-1}))$	$\widetilde{\mathcal{O}}(\epsilon^{-1}p_0^{-2})$
QPE with semi-classical Fourier transform	$\widetilde{\mathcal{O}}(\epsilon^{-1}p_0^{-1})$	$\widetilde{\mathcal{O}}(p_0^{-1}\mathrm{polylog}(\epsilon^{-1}))$	$\widetilde{\mathcal{O}}(\epsilon^{-1}p_0^{-2})$
QEEA [149]	$\widetilde{\mathcal{O}}(\epsilon^{-1}\mathrm{polylog}(p_0^{-1}))$	$\widetilde{\mathcal{O}}(\epsilon^{-3}p_0^{-2})$	$\widetilde{\mathcal{O}}(\epsilon^{-4}p_0^{-2})$

Table 3.2: Comparison of the maximal evolution time, the number of repetitions (the number of times we need to run the quantum circuit), and the total evolution time needed for estimating the ground state energy to within error ϵ, using the three methods that require only one ancilla qubit: the method in this work, QPE with semi-classical Fourier transform that uses only one ancilla qubit, and the QEEA in Ref. [149]. The overlap between the initial state and the ground state is assumed to be p_0. The number of repetitions is also the number of times we need to prepare the initial state. An analysis of the QEEA in Ref. [149] can be found in Appendix 3.9.

However, if we can only perform time evolution through Hamiltonian simulation, then these relations become more complicated. If advanced Hamiltonian simulation methods [115, 114, 34] can be used, the additional cost would be asymptotically negligible, since to ensure an ϵ' error for time evolution for time T the cost is $\mathcal{O}(T\mathrm{polylog}(T\epsilon'^{-1}))$. Hence the cost is only worse than that in the ideal case by a poly-logarithmic factor. However, for early fault-tolerant quantum computers, as discussed in Refs. [103, 47], Trotter formulas [157] are generally favored. Running time evolution for time T with error at most ϵ' would entail a runtime of $\mathcal{O}(T^{1+1/p}\epsilon'^{-1/p})$. The additional cost will therefore prevent us from reaching the Heisenberg limit, though high-order Trotter formulas (i.e. with a large p) can allow us to get arbitrarily close to the Heisenberg limit. If one does not insist on having a Heisenberg-limited scaling, then randomized algorithms [46, 36, 52] may lead to lower gate count when only low precision is required.

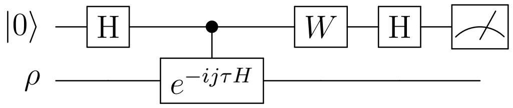
In Appendix 3.10 we analyze the circuit depth and the total runtime of our algorithm with time evolution performed using Trotter formulas. We also compare with QPE based on Trotter formulas. We found that when using Trotter formulas, our method has some additional advantage over QPE, achieving a polynomially better dependence on p_0 (i.e. η in Appendix 3.10) in the total runtime. The total runtime scales like $\epsilon^{-1-o(1)}$ using our algorithm with Trotter formulas, and this only approximately reaches the Heisenberg limit ϵ^{-1} in terms of the total runtime. However, it is worth noting that none of the other methods can strictly reach the Heisenberg limit using Trotter formulas. Otherwise we can instead perform Hamiltonian simulation with the exponentially accurate methods to go below the Heisenberg limit, which is an impossible task. Despite the sub-optimal asymptotic scaling, with tight error analysis [60, 163, 55, 177] Trotter formulae may outperform the advanced Hamiltonian simulation techniques discussed above in terms of the gate complexity, especially when only moderate accuracy is needed.

Organization

The rest of the paper is organized as follows. In Section 3.2 we introduce the quantum circuit we are going to use, and introduce the CDF which is going to play an important role in our algorithm, and give an overview of the ground state energy estimation algorithm. In Section 3.3 we discuss how to approximate the CDF. In Section 3.4 we show that the ground state energy can be estimated by inverting the CDF, and present the complexity of our algorithm (Corollary 15). In Section 3.5 we present the details of our algorithm for post-processing the measurement data and analyze the complexity.

3.2 Overview of the method

We want to keep the quantum circuit we use as simple as possible. In this work we use the following circuit



(3.1)

where H is the Hadamard gate. We choose $W = I$ or $W = S^{\dagger}$ where S is the phase gate, depending on the quantity we want to estimate. The quantum circuit is simple and uses only one ancilla qubit as required. The quantum circuit itself has been used in previous methods [102, 149]. However, our algorithm uses a different strategy for querying the circuit and for classical post-processing, and results in lower total evolution time and/or maximal evolution time achieving the goals (1) and (3) listed on Page 46.

This circuit requires controlled time evolution, which can be non-trivial to implement. The idea of removing controlled operation in phase estimation has also been considered in [40]. Here we can use ideas from Refs. [96, 118, 142, 127] to remove the need to perform controlled time evolution. But this type of approach requires an eigenstate of H with known eigenvalue that is easy to prepare. In a second-quantized setting we can simply use the vacuum state. We will discuss this in detail in Appendix 3.11.

Using the circuit in (3.1), in order to estimate $\operatorname{Re}\operatorname{Tr}[\rho e^{-ij\tau H}]$, where j is an arbitrary integer and τ is a real number, we set $W = I$. We introduce a random variable X_j and set it to be 1 when the measurement outcome is 0, and -1 when the measurement outcome is 1. Then

$$\mathbb{E}[X_j] = \operatorname{Re}\operatorname{Tr}[\rho e^{-ij\tau H}]. \tag{3.2}$$

Similarly for $\operatorname{Im}\operatorname{Tr}[\rho e^{-ij\tau H}]$, we set $W = S^\dagger$, and introduce a random variable Y_j that depends in the same way on the measurement outcome. We have

$$\mathbb{E}[Y_j] = \operatorname{Im}\operatorname{Tr}[\rho e^{-ij\tau H}]. \tag{3.3}$$

The parameter τ is chosen to normalize the Hamiltonian. Specifically, we choose τ so that $\tau\|H\| < \pi/3$. We remark that τ should be chosen to be $\mathcal{O}(\|H\|^{-1})$, and to avoid unnecessary overheads we want its scaling to be as close to $\Theta(\|H\|^{-1})$ as possible.

We can define a spectral measure of τH associated with ρ. The spectral measure is

$$p(x) = \sum_{k=0}^{K-1} p_k \delta(x - \tau\lambda_k), \quad x \in [-\pi, \pi]. \tag{3.4}$$

Here K is the number of different eigenvalues, λ_k's are the distinct eigenvalues arranged in ascending order, and each p_k is the corresponding overlap, as defined in the Introduction. We extend it to a 2π-periodic function by $p(x+2\pi) = p(x)$ so that the Fourier transform can be performed on the interval $[0, 2\pi]$ instead of the whole real line, which leads to a discrete Fourier spectrum. Note that because of the assumption $\tau\|H\| < \pi/3$, within the interval $[-\pi, \pi]$, $p(x)$ is supported in $(-\pi/3, \pi/3)$. Next we consider the cumulative distribution function (CDF) associated with this measure.

We define the 2π-periodic Heaviside function by

$$H(x) = \begin{cases} 1, & x \in [2k\pi, (2k+1)\pi), \\ 0, & x \in [(2k-1)\pi, 2k\pi), \end{cases} \tag{3.5}$$

where $k \in \mathbb{Z}$. The CDF is usually defined by $C(x) = \sum_{k:\lambda_k \le x} p_k$. This is however not a 2π-periodic function and thus will create technical difficulties in later discussions.

Therefore instead of the usual definition, we define

$$C(x) = (H * p)(x), \tag{3.6}$$

where $*$ denotes convolution. There is ambiguity at the jump discontinuities, and we define the values of $C(x)$ at these points by requiring $C(x)$ to be right-continuous. We check that this definition agrees with the usual definition when $x \in (-\pi/3, \pi/3)$, which is the interval that contains all the eigenvalues of τH:

$$\begin{aligned} C(x) &= \int_{-\pi}^{\pi} H(y)p(x-y)\mathrm{d}y = \int_{0}^{\pi} p(x-y)\mathrm{d}y \\ &= \int_{x-\pi}^{x} p(y)\mathrm{d}y = \int_{-\pi}^{x} p(y)\mathrm{d}y = \sum_{k:\lambda_k \le x} p_k. \end{aligned}$$

Consequently $C(x)$ is a right-continuous non-decreasing function in $(-\pi/3, \pi/3)$.

If we could evaluate the CDF then we would be able to locate the ground state energy. This is because the CDF is a piecewise constant function. Each of its jumps in the interval $(-\pi/3, \pi/3)$ corresponds to an eigenvalue of τH. In order to find the ground state energy we only need to find where $C(x)$ jumps from zero to a non-zero value. However, in practice we cannot evaluate the CDF exactly. We will see that we are able to approximate, in a certain sense as will be made clear later, the CDF using a function we call the approximate CDF (ACDF). To this end we first define an approximate Heaviside function $F(x) = \sum_{|j|\le d} \hat{F}_j e^{ijx}$ such that

$$|F(x) - H(x)| \le \epsilon, \quad x \in [-\pi+\delta, -\delta] \cup [\delta, \pi-\delta]. \tag{3.7}$$

The construction of this function is provided in Lemma 18, where $\hat{F}_j$ is written as $\hat{F}_{d,\delta,j}$. Here the parameters d and δ need to be chosen to control the accuracy of this approximation, and their choices will be discussed later. We also omit the d and δ dependence in the subscripts for simplicity. With this $F(x)$ we define the ACDF by

$$\widetilde{C}(x) = (F * p)(x). \tag{3.8}$$

In Section 3.3 we will discuss how to evaluate this ACDF using the circuit in (3.1). The ACDF and CDF are related through the following inequalities

$$C(x-\delta) - \epsilon \le \widetilde{C}(x) \le C(x+\delta) + \epsilon \tag{3.9}$$

for any $|x| \le \pi/3$, $0 < \delta < \pi/6$ and $\epsilon > 0$. We prove these inequalities in Appendix 3.8. Given the statistical estimation of the ACDF $\widetilde{C}(x)$, these inequalities

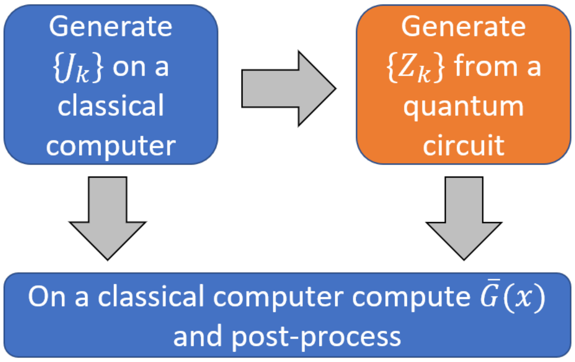

Generate $\{J_k\}$ on a classical computer
Generate $\{Z_k\}$ from a quantum circuit
On a classical computer compute $\bar{G}(x)$ and post-process

Figure 3.2: An illustration of the classical and quantum components of our algorithm: (1) generate samples $\{J_k\}$ from (3.11); (2) use $\{J_k\}$ to generate $\{Z_k\}$ according to (3.13); (3) compute $\bar{G}(x)$ through (3.16). The ground state energy estimate can be obtained through post-processing as discussed in Section 3.4. Only Step (2) needs to be performed on a quantum computer.

enable us to estimate where the jumps of the CDF occur, which leads to an estimate of the ground state energy.

By approximately evaluating the ACDF $\widetilde{C}(x)$ for certain chosen x, and through eq. (3.9), we can perform a binary search to locate the ground state energy in smaller and smaller intervals. The algorithm to do this and the total computational cost required to estimate the ground state energy to precision ϵ at a confidence level $1-\vartheta$ are discussed in Sections 3.4 and 3.5.

3.3 Evaluating the ACDF

In this section we discuss how to evaluate the ACDF $\widetilde{C}(x)$. We first expand it in the following way:

$$\begin{aligned} \widetilde{C}(x) &= \sum_{|j|\leq d} \hat{F}_j \int_{-\pi}^{\pi} p(y) e^{ij(x-y)} \mathrm{d}y \\ &= \sum_{|j|\leq d} \hat{F}_j e^{ijx} \mathrm{Tr}[\rho e^{-ij\tau H}], \end{aligned} \tag{3.10}$$

where the spectral measure $p(x)$ is defined in (3.4). In going from the first line to the second line in the above equation we have used the fact that

$$\int_{-\pi}^{\pi} p(y)e^{-ijy}\mathrm{d}y = \sum_{k=0}^{K-1} \mathrm{Tr}[\rho\Pi_k]e^{-ij\tau\lambda_k} = \mathrm{Tr}[\rho e^{-ij\tau H}].$$

One might want to evaluate each $\mathrm{Tr}[\rho e^{-ij\tau H}]$ using Monte Carlo sampling since this quantity is equal to $\mathbb{E}[X_j + iY_j]$. If we want to evaluate all $\mathrm{Tr}[\rho e^{-ij\tau H}]$ to any accuracy at all, we need to sample each X_j and Y_j at least once. Then the total evolution time is is at least $\tau\sum_{|j|\leq d}|j| = \Omega(\tau d^2)$. Later we will see we need to choose $d = \mathcal{O}(\epsilon^{-1}\mathrm{polylog}(\epsilon^{-1}p_0^{-1}))$ to ensure the ground state energy estimate has an additive error smaller than ϵ. Hence this total evolution time would give rise to a ϵ^{-2} dependence in the runtime.

In order to avoid this ϵ^{-2} dependence, instead of evaluating all the terms we stochastically evaluate (3.10) as a whole. The idea we are going to describe is inspired by the unbiased version of the multi-level Monte Carlo method [140, 141]. We define a random variable J that is drawn from $\{-d, -d+1, \ldots, d\}$, with probability

$$\Pr[J = j] = \frac{|\hat{F}_j|}{\mathcal{F}}, \tag{3.11}$$

where the normalization factor $\mathcal{F} = \sum_{|j|\leq d}|\hat{F}_j|$. We let θ_j be the argument of $\hat{F}_j$, i.e. $\hat{F}_j = |\hat{F}_j|e^{i\theta_j}$. Then

$$\begin{aligned}\mathbb{E}[(X_J + iY_J)e^{i(\theta_J+Jx)}] &= \sum_{|j|\leq d}\mathbb{E}[X_j + iY_j]e^{i(\theta_j+jx)}\Pr[J=j]\\ &= \frac{1}{\mathcal{F}}\sum_{|j|\leq d}\mathrm{Tr}[\rho e^{-ij\tau H}]e^{ijx}\hat{F}_j\\ &= \frac{\widetilde{C}(x)}{\mathcal{F}},\end{aligned} \tag{3.12}$$

where we have used (3.2) and (3.3). For simplicity we write X_J and Y_J into a complex random variable

$$Z = X_J + iY_J \in \{\pm 1 \pm i\}. \tag{3.13}$$

Therefore we can use

$$G(x; J, Z) = \mathcal{F}Ze^{i(\theta_J+Jx)} \tag{3.14}$$

as an unbiased estimate of $\widetilde{C}(x)$. The variance can be bounded by:

$$\mathrm{var}[G(x)] \leq \mathcal{F}^2\mathbb{E}[|X_J|^2 + |Y_J|^2] \leq 2\mathcal{F}^2. \tag{3.15}$$

Here we have used the fact that $|X_j|, |Y_j| \leq 1$.

From the above analysis, we can generate N_s independent samples of (J, Z), denoted by (J_k, Z_k), $k = 1, 2, \ldots, N_s$, and then take the average

$$\bar{G}(x) = \frac{1}{N_s} \sum_{k=1}^{N_s} G(x; J_k, Z_k), \tag{3.16}$$

which can be used to estimate $\widetilde{C}(x)$ in an unbiased manner. The variance is upper bounded by $2\mathcal{F}^2/N_s$. In order to make the variance upper bounded by a given σ^2, we need $N_s = \mathcal{O}(\mathcal{F}^2/\sigma^2)$. The expected total evolution time is

$$N_s \tau \mathbb{E}[|J|] = \frac{\mathcal{F}\tau}{\sigma^2} \sum_{|j| \leq d} |\hat{F}_j||j|.$$

Furthermore, by Lemma 18 (iii) we have $|\hat{F}_j| \leq C|j|^{-1}$ for some constant C. Therefore

$$\mathcal{F} = \mathcal{O}(\log(d)), \quad \sum_{|j| \leq d} |\hat{F}_j||j| = \mathcal{O}(d).$$

The number of samples and the expected total evolution time are therefore

$$N_s = \mathcal{O}\left(\frac{\log^2(d)}{\sigma^2}\right), \quad N_s \tau \mathbb{E}[|J|] = \mathcal{O}\left(\frac{\tau d \log(d)}{\sigma^2}\right), \tag{3.17}$$

respectively. We can see that in this way we have avoided the d^2 dependence, which shows up in a term-by-term evaluation.

In Figure 3.3 we show the plot of the ACDF obtained through our method for the Fermi-Hubbard model. The details on this numerical experiment can be found in Appendix 3.12. We can estimate the ground state energy from the ACDF in a heuristic manner: we let

$$x^\star = \inf\{x : \bar{G}(x) \geq \eta/2\},$$

and $x^\star/\tau$ is an estimate for the ground state energy λ_0. Here η is chosen so that $p_0 \geq \eta$. In Section 3.5 we describe a more elaborate method to achieve the prescribed accuracy and confidence level. However, this heuristic method seems to work reasonably well in practice. In Figure 3.4 we show the scaling of the ground state energy estimation error, the total evolution time, and the maximal evolution time, with respect to $\delta = \tau\epsilon$ (δ here is the parameter needed to construct $\{\hat{F}_j\}$ using Lemma 18), where ϵ is the allowed error. Both the total evolution time and the maximal evolution time are proportional to ϵ^{-1}. The details on this numerical experiment can also be found in Appendix 3.12.

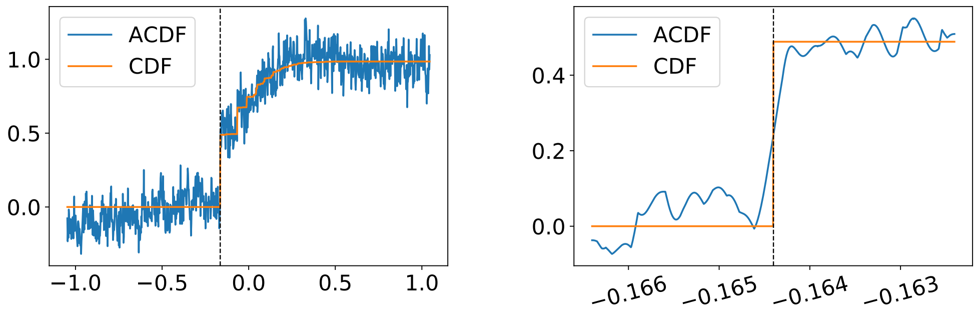


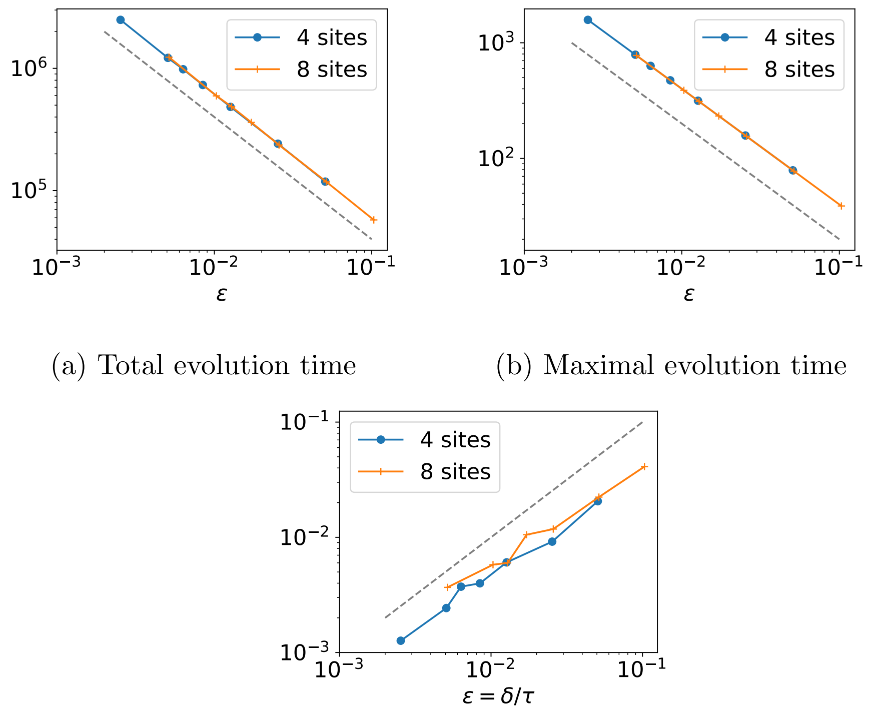
Figure 3.3: $\bar{G}(x)$ and the CDF $C(x)$, for $x \in [-\pi/3, \pi/3]$ (left) and the zoom in view around $\tau\lambda_0$ (right), the ground state energy for τH where H is the Hamiltonian for the 8-site Hubbard model with $U/t = 4$ at half-filling. The dashed vertical line is $x = \tau\lambda_0$. The parameters are $\delta = 2 \times 10^{-4}$, $d = 2 \times 10^4$, $\tau = \pi/(4\|H\|)$. In total 3000 samples are used.

3.4 Estimating the ground state energy

In this section we discuss how to estimate the ground state energy with guaranteed error bound and confidence level from the samples generated on classical and quantum circuits discussed in Sections 3.2 and 3.3. First we note that the CDF $C(x) = 0$ for all $-\pi/3 < x < \tau\lambda_0$, and $C(x) > 0$ for all $\tau\lambda_0 \leq x < \pi/3$. Therefore getting the ground state energy out of the CDF can be seen as inverting the CDF: we only need to find the smallest x such that $C(x) > 0$. One might consider performing a binary search to find such a point, but we run into a problem immediately: we only have access to estimates of $C(x)$ with statistical noise, and we cannot tell if the estimate is greater than zero is due to $C(x) > 0$ or is merely due to statistical noise. We therefore need to make the search criterion more robust to noise.

Note that the CDF cannot take values between 0 and p_0: $C(x) \geq p_0$ for $\tau\lambda_0 \leq x < \pi/3$ and $C(x) = 0$ for $-\pi/3 < x < \tau\lambda_0$. Now suppose we know $p_0 \geq \eta$, then for any x, rather than distinguishing between $C(x) = 0$ and $C(x) > 0$, we instead distinguish between $C(x) = 0$ and $C(x) \geq \eta/2$ (here $\eta/4$ is chosen to be consistent with later discussion and it can be any number between 0 and 1 times η). In this setting, if the estimate of $C(x)$ is larger than $\eta/4$ then we tend to believe that $C(x) \geq \eta/2$, and if the estimate is smaller than $\eta/4$ then we tend to believe that $C(x) = 0$. Thus we can tolerate an error that is smaller than $\eta/4$.

It may appear that we can find the ground state energy by performing a binary search for the point at which $C(x)$ first becomes larger than $\eta/2$. However, we can



(a) Total evolution time

(b) Maximal evolution time

(c) Average energy estimation error

Figure 3.4: The total evolution time (a), maximal evolution time (b), and the average ground state energy estimation error (c), for 4-site and 8-site Hubbard model with $U/t = 4$ at half-filling. The horizontal axis is the error threshold $\epsilon = \delta/\tau$. In (a) and (b) the grey dash lines have slope -1, and in (c) the grey dashed line (with slope 1) shows the value of ϵ. For each δ, d is chosen to be $d = 4/\delta$, with 1800 samples, and $\tau = \pi/(4\|H\|)$. The maximal evolution time is $\tau d = 4\tau/\delta$.

only estimate the continuous function $\widetilde{C}(x)$, which cannot uniformly approximate $C(x)$. This is because $C(x)$ has many jump discontinuities (each of which corresponds to an eigenvalue). As a result, we cannot perform this binary search procedure directly.

From the above discussion we need a search criterion that can be checked via $\widetilde{C}(x)$. We consider the following criterion:

Problem 13 (Inverting the CDF)**.** *For* $0 < \delta < \pi/6$*,* $0 < \eta < 1$*, find* $x^\star \in$

$(-\pi/3,\pi/3)$ such that

$$C(x^\star+\delta)>\eta/2,\quad C(x^\star-\delta)<\eta. \tag{3.18}$$

Firstly we verify that this can be checked via $\widetilde{C}(x)$. In (3.9), if we choose $x=x^\star$, $\epsilon=\eta/6$, then $\widetilde{C}(x^\star)>(2/3)\eta$ implies $C(x^\star)>\eta/2$, and $\widetilde{C}(x^\star)<(5/6)\eta$ implies $C(x^\star)<\eta$. Therefore we only need to find $x^\star$ satisfying $(2/3)\eta<\widetilde{C}(x^\star)<(5/6)\eta$ to satisfy this criterion. Secondly we show that an $x^\star$ satisfying this criterion gives us an estimate of the ground state energy to within additive error δ/τ. Suppose we choose $\eta>0$ so that $p_0\geq\eta$. Then if we solve Problem 13 we will find an $x^\star$ such that $C(x^\star+\delta)>\eta/2>0$ and $C(x^\star-\delta)<\eta\leq p_0$. $C(x^\star+\delta)>0$ indicates that $x^\star+\delta\geq\tau\lambda_0$. Since $C(x)$ cannot take value between 0 and p_0, $C(x^\star-\delta)<p_0$ indicates $C(x^\star-\delta)=0$ and thus $x^\star-\delta<\tau\lambda_0$. Hence we know $|x^\star-\tau\lambda_0|\leq\delta$. If we choose $\delta=\tau\epsilon$ and $\widetilde{\lambda}_0=x^\star/\tau$, then

$$|\widetilde{\lambda}_0-\lambda_0|\leq\epsilon.$$

Then $\widetilde{\lambda}_0$ is our desired estimate.

Note that (3.18) is a weaker requirement than $\eta/2<C(x^\star)<\eta$, for which due to the discontinuity of $C(x)$ the required $x^\star$ may not exist. However an $x^\star$ satisfying (3.18) must exist. In fact, let $a=\sup\{x\in(-\pi/3,\pi/3):C(x)\leq\eta/2\}$ and $b=\inf\{x\in(-\pi/3,\pi/3):C(x)\geq\eta\}$. Then because $C(x)$ is monotonously increasing, $a\leq b$, and any $x^\star\in[a-\delta,b+\delta)$ satisfies (3.18).

Using the samples $\{J_k\}$ and $\{Z_k\}$ generated on classical and quantum circuits respectively, we are able to solve Problem 13.

Theorem 14 (Inverting the CDF)**.** *With samples $\{J_k\}_{k=1}^M$ satisfying $|J_k|\leq d$ and $\{Z_k\}_{k=1}^M$, generated according to (3.11) and (3.13) respectively, we can solve Problem 13 on a classical computer with probability at least $1-\vartheta$, for $d=\mathcal{O}(\delta^{-1}\log(\delta^{-1}\eta^{-1}))$ and $M=\mathcal{O}(\eta^{-2}\log^2(d)(\log\log(\delta^{-1})+\log(\vartheta^{-1})))$. The classical post-processing cost is*

$$\widetilde{\mathcal{O}}(\eta^{-2}\log^3(\delta^{-1})\log(\vartheta^{-1})). \tag{3.19}$$

To generate the samples $\{Z_k\}_{k=1}^M$ on a quantum circuit, the expected total evolution time and the maximal evolution time are

$$\tau M\mathbb{E}[|J|]=\widetilde{\mathcal{O}}(\tau\delta^{-1}\eta^{-2}\log(\vartheta^{-1})), \tag{3.20}$$

and

$$\tau d=\mathcal{O}\left(\tau\delta^{-1}\log(\delta^{-1}\eta^{-1})\right). \tag{3.21}$$

respectively.

We will prove this theorem by constructing the algorithm for classical post-processing in Section 3.5. Since solving Problem 13 enables us to estimate the ground state energy as discussed above, from Theorem 14 we have the following corollary:

Corollary 15 (Ground state energy)**.** *With samples $\{J_k\}_{k=1}^M$ satisfying $|J_k| \leq d$ and $\{Z_k\}_{k=1}^M$, generated according to* (3.11) *and* (3.13) *respectively, we can estimate the ground state energy λ_0 to within additive error ϵ on a classical computer with probability at least $1-\vartheta$, if $p_0 \geq \eta$ for some known η, $d = \mathcal{O}(\epsilon^{-1}\tau^{-1}\log(\epsilon^{-1}\tau^{-1}\eta^{-1}))$, and $M = \mathcal{O}(\eta^{-2}\log^2(d)(\log\log(\epsilon^{-1}\tau^{-1}) + \log(\vartheta^{-1})))$. The classical post-processing cost is $\mathcal{O}(\eta^{-2}\mathrm{polylog}(\epsilon^{-1}\tau^{-1}\eta^{-1}))$. The expected total evolution time and the maximal evolution time are $\mathcal{O}(\epsilon^{-1}\eta^{-2}\mathrm{polylog}(\epsilon^{-1}\tau^{-1}\eta^{-1}))$ and $\mathcal{O}(\epsilon^{-1}\mathrm{polylog}(\epsilon^{-1}\tau^{-1}\eta^{-1}))$ respectively.*

Usually the Heisenberg limit is defined in terms of the root-mean-square error (RMSE) of the estimate. In this paper we focus on ensuring the error of the ground state energy to be below a threshold ϵ with probability at least $1-\vartheta$. From Corollary 15, our algorithm only has a logarithmic dependence on ϑ^{-1}, and the error can be at most $2\|H\|$, we can easily ensure the RMSE is $\mathcal{O}(\epsilon)$ using the result by choosing $\vartheta = \mathcal{O}(\epsilon^2\|H\|^{-2})$. We can see the total evolution time scaling with respect to ϵ is still $\widetilde{\mathcal{O}}(\epsilon^{-1})$.

Remark 16 (System size dependence)**.** *One might notice the absence of an explicit system size dependence in the evolution time scaling in Theorem 14 and Corollary 15. This is because, as mentioned before in the Introduction, the total evolution time depends on the system size indirectly through two parameters τ and η. Moreover, if we consider the dependence of the total runtime on the system size, we also need to account for the overhead that comes from performing Hamiltonian simulation. This overhead and the scaling of η with respect to the system size are highly problem-specific and are independent from the tasks we are considering in this paper, and hence we will not discuss them in more detail. Because the Hamiltonian norm can generally be upper bounded by a polynomial of the system size, and the total evolution time dependence on τ^{-1} is poly-logarithmic, τ contributes a poly-logarithmic overhead in the system size dependence.*

3.5 Inverting the CDF

In this section we prove Theorem 14 by constructing the classical post-processing algorithm to solve Problem 13 using samples from a quantum circuit. Since we want to search for an $x^\star$ satisfying the requirement (3.18), a natural idea is to use binary search. Our setting is somewhat different from the usual binary search setting, but we will show that a similar approach still works. The current setting differs from the setting of binary search mainly in two ways: first any $x^\star \in [\tau\lambda_0 - \delta, \tau\lambda_0 + \delta]$ satisfies the requirement (3.18) and can therefore be a target. When performing binary search we want to be able to tell if the target is to the left or right of a given x, but here the targets may be on both sides of x. When this happens there is some uncertainty as to how the algorithm will proceed next. However in our algorithm we will show that this does not present a problem. Also, because this algorithm is based on random samples, there is some failure probability in each search step. We will use a majority voting procedure to suppress the failure probability so that in the end the algorithm will produce a correct answer with probability arbitrarily close to 1.

We suppose we are given independent samples of (J, Z) defined in (3.11) and (3.13) generated from a quantum circuit. We denote these samples by $\{(J_k, Z_k)\}_{k=1}^M$. We divide them into N_b batches of size N_s, where $N_s N_b = M$. This division is for the majority voting procedure we mentioned above. The maximal evolution time needed to generate these samples is proportional to $\max_k |J_k| \leq d$. The expected total evolution time we will need is proportional to $M\mathbb{E}[|J|]$.

We first reduce Problem 13 into a decision problem. For any $x \in (-\pi/3, \pi/3)$, one of the following must be true:

$$C(x+\delta) > \eta/2, \quad \text{or} \quad C(x-\delta) < \eta. \tag{3.22}$$

If there is a subroutine that tells us which one of the two is correct, or randomly picks one when both are correct, then we can use it to find $x^\star$. We assume such a subroutine, which uses $\{(J_k, Z_k)\}_{k=1}^M$, exists and denote it by the name $\texttt{CERTIFY}(x, \delta, \eta, \{(J_k, Z_k)\})$. The subroutine returns either 0 or 1: 0 for $C(x+\delta) > \eta/2$ being true, and 1 for $C(x-\delta) < \eta$ being true.

In Algorithm 2, with $\texttt{CERTIFY}(x, \delta, \eta, \{(J_k, Z_k)\})$, we describe the algorithm to solve Problem 13. This algorithm we denote by $\texttt{INVERT\_CDF}(\delta, \eta, \{(J_k, Z_k)\})$. It runs as follows: we start with $x_{0,0} = -\pi/3$ and $x_{1,0} = \pi/3$. They are chosen so that $C(x_{1,0}) > \eta/2$ and $C(x_{0,0}) < \eta$. Let ℓ be the number of iterations we have performed, and $\ell = 0$ at the beginning. At each iteration, we let $x_\ell = (x_{0,\ell} + x_{1,\ell})/2$, and run $\texttt{CERTIFY}(x_\ell, (2/3)\delta, \eta, \{(J_k, Z_k)\})$. This tells us either $C(x_\ell + (2/3)\delta) > \eta/2$ or $C(x_\ell - (2/3)\delta) < \eta$. If the former then we let $x_{0,\ell+1} = x_{0,\ell}$, $x_{1,\ell+1} = x_\ell + (2/3)\delta$, and if the latter we let $x_{0,\ell+1} = x_\ell + (2/3)\delta$, $x_{1,\ell+1} = x_{1,\ell}$. This is done so that for each ℓ we have

$$C(x_{0,\ell}) < \eta, \quad C(x_{1,\ell}) > \eta/2. \tag{3.23}$$

We then let $\ell \leftarrow \ell + 1$ and go to the next iteration. The algorithm stops once $x_{1,\ell} - x_{0,\ell} \leq 2\delta$. We denote the total number of iterations by L. The output is

$x_L = (x_{0,L} + x_{1,L})/2$. Because (3.23) holds for each iteration we have

$$C(x_L - \delta) \leq C(x_{0,L}) < \eta, \quad C(x_L + \delta) \geq C(x_{1,L}) > \eta/2.$$

Thus we can see x_L satisfies the requirements for $x^\star$ in Problem 13. The next question is, how many iterations does it take to satisfy the stopping criterion? Regardless of the outcome of the CERTIFY subroutine, we always have

$$x_{1,\ell+1} - x_{0,\ell+1} = \frac{1}{2}(x_{1,\ell} - x_{0,\ell}) + \frac{2}{3}\delta.$$

From this we can see

$$x_{1,\ell} - x_{0,\ell} = \frac{2\pi/3 - (4/3)\delta}{2^\ell} + \frac{4}{3}\delta.$$

Therefore it takes $L = \mathcal{O}(\log(\delta^{-1}))$ iterations for the algorithm to stop.

Algorithm 2 INVERT_CDF

```
Require: $\delta, \eta, \{(J_k, Z_k)\}$
  $x_0 \leftarrow -\pi/3$, $x_1 \leftarrow \pi/3$;
  while $x_1 - x_0 > 2\delta$ do
    $x \leftarrow (x_0 + x_1)/2$;
    $u \leftarrow$ CERTIFY$(x, (2/3)\delta, \eta, \{(J_k, Z_k)\})$;
    if $u = 0$ then
      $x_1 \leftarrow x + (2/3)\delta$;
    else
      $x_0 \leftarrow x - (2/3)\delta$;
    end if
  end while
Ensure: $(x_0 + x_1)/2$
```

Next we discuss how to construct the subroutine CERTIFY$(x, \delta, \eta, \{(J_k, Z_k)\})$. While we cannot directly evaluate the CDF $C(x)$ for any x, we can estimate the ACDF $\widetilde{C}(x)$ using the data $\{J_k\}$ and $\{Z_k\}$. We can let $\epsilon = \eta/8$ in (3.7) and choose $d = \mathcal{O}(\delta^{-1}\log(\delta^{-1}\eta^{-1}))$ according to Lemma 18. Then by (3.9), we have $C(x-\delta) \leq \widetilde{C}(x) + \eta/8$ and $C(x+\delta) \geq \widetilde{C}(x) - \eta/8$. One of the following must be true:

$$\widetilde{C}(x) > (5/8)\eta, \quad \text{or} \quad \widetilde{C}(x) < (7/8)\eta, \tag{3.24}$$

then the former implies $C(x+\delta) > \eta/2$ and the latter $C(x-\delta) < \eta$. Therefore the CERTIFY subroutine only needs to decide which one of the two is correct or to output a random choice when both are correct.

As discussed in Section 3.3, $\bar{G}(x)$ is an unbiased estimate of $\widetilde{C}(x)$. We use $\{J_k\}$ and $\{Z_k\}$ to get N_b samples for $\bar{G}(x)$, denoted by $\bar{G}_r(x)$, via

$$\bar{G}_r(x) = \frac{1}{N_s}\sum_{k=1}^{N_s} G(x; J_{(r-1)N_s+k}, Z_{(r-1)N_s+k})$$

for $r = 1, 2, \ldots, N_b$. Here $G(x; J, Z)$ is defined in (3.14). For each r, we compare $\bar{G}_r(x)$ with $(3/4)\eta$. If $\bar{G}_r(x) > (3/4)\eta$ for a majority of batches, then we tend to believe $\widetilde{C}(x) > (5/8)\eta$ and output 0 for $C(x+\delta) > \eta/2$. Otherwise, we tend to believe $\widetilde{C}(x) < (7/8)\eta$ and output 1 for $C(x-\delta) < \eta$. This is the majority voting procedure we mentioned earlier. For the pseudocode for the subroutine see Algorithm 3.

Algorithm 3 `CERTIFY`

```
Require: $x, \delta, \eta, \{(J_k, Z_k)\}$
  $b \leftarrow 0$, $c \leftarrow 0$;
  for $r = 1, 2, \ldots, N_b$ do
    $\bar{G}_r(x) \leftarrow (1/N_s)\sum_{k=1}^{N_s} G(x; J_{(r-1)N_s+k}, Z_{(r-1)N_s+k})$; {$G(x; J, Z)$ defined in (3.14)}
    if $\bar{G}_r(x) > (3/4)\eta$ then
      $c \leftarrow c + 1$;
    end if
  end for
  if $c \leq B/2$ then
    $b \leftarrow 1$;
  end if
Ensure: $b$
```

In the `CERTIFY` subroutine, an error occurs when $\widetilde{C}(x) > (5/8)\eta$ yet a majority of estimates $\bar{G}_r(x)$ are smaller than $(3/4)\eta$, or when $\widetilde{C}(x) < (7/8)\eta$ yet a majority of estimates $\bar{G}_r(x)$ are larger than $(3/4)\eta$. We need to make the probability of this kind of error occurring upper bounded by ν. First we assume $\widetilde{C}(x) > (5/8)\eta$. Then for each r, by Markov's inequality, we have

$$\Pr[\bar{G}_r(x) < (3/4)\eta] \leq \frac{64\,\mathrm{var}[\bar{G}_r(x)]}{\eta^2}.$$

We want to make this probability at most $1/4$. Therefore we need $\mathrm{var}[\bar{G}_r(x)] \leq \eta^2/256$. To ensure this, by (3.17) in which we let $\sigma^2 = \eta^2/256$, we can choose

$$N_s = \mathcal{O}\left(\frac{\log^2(d)}{\eta^2}\right). \tag{3.25}$$

Then by the Chernoff bound the probability of the majority of estimates $\bar{G}_r(x)$ being smaller than $(3/4)\eta$ is at most $e^{-C'N_b}$ for some constant C'. In order to make this probability bounded by ν we only need to let $N_b = \mathcal{O}(\log(\nu^{-1}))$.

In the algorithm `INVERT_CDF`, the subroutine `CERTIFY` is used $L = \mathcal{O}(\log(\delta^{-1}))$ times. If an error occurs in a single run of `CERTIFY` with probability at most ν then in the total L times we use this subroutine the probability of an error occurring is at most $L\nu$. Therefore in order to ensure that an error occurs with probability at most ϑ in `INVERT_CDF`, we need to set $\nu = \vartheta/L$. Therefore $N_b = \mathcal{O}(\log(L\vartheta^{-1})) = \mathcal{O}(\log\log(\delta^{-1}) + \log(\vartheta^{-1}))$.

The above analysis shows that in order to solve Problem 13 the total evolution time is $M\mathbb{E}[|J|] = N_b N_s \mathbb{E}[|J|]$. We evaluate $N_s\mathbb{E}[|J|]$ by (3.17) in which we let $\sigma^2 = \eta^2/256$ as discussed before when we estimate how large N_s needs to be in (3.25). Multiplying this by N_b we have (3.20). Note here we do not need to multiply by L because in each `CERTIFY` subroutine we can reuse the same $\{J_k\}$, $\{Z_k\}$. The maximal evolution time required is τd and this leads to (3.21). The main cost in classical post-processing comes from evaluating $\bar{G}_r(x)$. This needs to be done LN_b times. Each evaluation involves $\mathcal{O}(N_s) = \mathcal{O}(\eta^{-2}\log^2(d))$ arithmetic operations. The total runtime for classical post-processing is therefore $LN_bN_s = LM$, which leads to (3.19). Thus we have obtained all the cost estimates in Theorem 14 and proved the theorem.

3.6 Discussions

In this paper we presented an algorithm to estimate the ground state energy with Heisenberg-limited precision scaling. The quantum circuit we used requires only one ancilla qubit, and the maximal evolution time needed per run has a poly-logarithmic dependence on the overlap p_0. Such dependence on p_0 is exponentially better than that required by QPE using a similarly structured circuit using semi-classical Fourier transform, as discussed in Section 3.1. Both rigorous analysis and numerical experiments are done to validate the correctness and efficiency of our algorithm.

Although our algorithm has a near-optimal dependence on the precision, the dependence on p_0 (more precisely, on its lower bound η), which scales as p_0^{-2} in Corollary 15, is far from optimal compared to the $p_0^{-1/2}$ scaling in Refs. [81, 111]. Whether one can achieve this $p_0^{-1/2}$ scaling without using a quantum circuit with substantially larger maximal evolution time, and without using such techniques as LCU or block-encoding, remains an open question.

The probabilistic choice of the simulation time according to eq. (3.11) plays an important role in reducing the total evolution time. However, we may partially derandomize the algorithm following the spirit of the multilevel Monte Carlo (MLMC) method [82] in the classical setting. The method we developed for computing the approximate CDF in Section 3.3 is in fact a quite general approach for evaluating expectation values from matrix functions. This method can act as a substitute of the LCU method in many cases, especially in a near-term setting. Using this method to compute other properties of the spectrum, such as the spectral density, is a direction for future work.

3.7 Constructing the approximate Heaviside function

In this appendix we construct the approximate Heaviside function satisfying the requirement in (3.7). We need to first construct a smeared Dirac function, which we will use as a mollifier in constructing the approximate Heaviside function. To our best knowledge this particular version of smeared Dirac function has not been proposed in previous works.

Lemma 17. *We define $M_{d,\delta}(x)$ by*

$$M_{d,\delta}(x) = \frac{1}{\mathcal{N}_{d,\delta}} T_d\left(1 + 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}\right),$$

where $T_d(x)$ is the d-th Chebyshev polynomial of the first kind, and

$$\mathcal{N}_{d,\delta} = \int_{-\pi}^{\pi} T_d\left(1 + 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}\right) \mathrm{d}x.$$

Then

(i) $|M_{d,\delta}(x)| \leq \frac{1}{\mathcal{N}_{d,\delta}}$ *for all* $x \in [-\pi, -\delta] \cup [\delta, \pi]$*, and* $M_{d,\delta}(x) \geq -\frac{1}{\mathcal{N}_{d,\delta}}$ *for all* $x \in \mathbb{R}$*.*

(ii) $\int_{-\pi}^{\pi} M_{d,\delta}(x)\mathrm{d}x = 1$, $1 \leq \int_{-\pi}^{\pi} |M_{d,\delta}(x)|\mathrm{d}x \leq 1 + \frac{4\pi}{\mathcal{N}_{d,\delta}}$.

(iii) When $\tan(\delta/2) \leq 1 - 1/\sqrt{2}$*, we have*

$$\mathcal{N}_{d,\delta} \geq C_1 e^{d\delta/\sqrt{2}} \sqrt{\frac{\delta}{d}} \operatorname{erf}(C_2\sqrt{d\delta})$$

for some constants C_1 and C_2 that do not depend on d or δ.

Proof. We first note that, by the property of Chebyshev polynomials, when $x \in [-\pi, -\delta] \cup [\delta, \pi]$, i.e. $\cos(x) \leq \cos(\delta)$, we have $\left|T_d\left(1 + 2\frac{\cos(x)-\cos(\delta)}{1+\cos(\delta)}\right)\right| \leq 1$. This proves the first inequality in (i). Note that when $x \in [-\delta, \delta]$, $T_d\left(1 + 2\frac{\cos(x)-\cos(\delta)}{1+\cos(\delta)}\right) \geq -1$. Combine this and the first inequality with the fact that $M_{d,\delta}(x)$ is 2π-periodic we prove the second inequality in (i).

The first part of (ii) is obvious because of the definition of $\mathcal{N}_{d,\delta}$. For the second part, we have $\int_{-\pi}^{\pi} |M_{d,\delta}(x)|\mathrm{d}x \geq \int_{-\pi}^{\pi} M_{d,\delta}(x)\mathrm{d}x = 1$. Also

$$\begin{aligned}\int_{-\pi}^{\pi} |M_{d,\delta}(x)|\mathrm{d}x &= \left(\int_{-\pi}^{-\delta} + \int_{\delta}^{\pi}\right) |M_{d,\delta}(x)|\mathrm{d}x + \int_{-\delta}^{\delta} M_{d,\delta}(x)\mathrm{d}x \\ &\leq \frac{4\pi}{\mathcal{N}_{d,\delta}} + \left(\int_{-\pi}^{-\delta} + \int_{\delta}^{\pi}\right) M_{d,\delta}(x)\mathrm{d}x + \int_{-\delta}^{\delta} M_{d,\delta}(x)\mathrm{d}x \qquad (3.26) \\ &= 1 + \frac{4\pi}{\mathcal{N}_{d,\delta}}.\end{aligned}$$

We now prove (iii). This requires lower bounding $T_d\left(1 + 2\frac{\cos(x)-\cos(\delta)}{1+\cos(\delta)}\right)$ when $x \in [-\delta, \delta]$. For δ small enough so that

$$\max_x 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)} = 2\tan^2(\delta/2) \leq 3 - \sqrt{2},$$

which is equivalent to $\tan(\delta/2) \leq 1 - 1/\sqrt{2}$, we can use [112, Lemma 13] to provide a lower bound for the $x \in [-\delta, \delta]$ case:

$$T_d\left(1 + 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}\right) \geq \frac{1}{2}\exp\left(\sqrt{2}d\sqrt{\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}}\right). \qquad (3.27)$$

By the elementary inequality $|\sin(x)| \leq |x|$, we have

$$\begin{aligned}\sqrt{\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}} &= \sqrt{\tan^2\left(\frac{\delta}{2}\right) - \frac{\sin^2(x/2)}{\cos^2(\delta/2)}} = \tan\left(\frac{\delta}{2}\right)\sqrt{1 - \frac{\sin^2(x/2)}{\sin^2(\delta/2)}} \\ &\geq \tan\left(\frac{\delta}{2}\right)\left(1 - \frac{\sin^2(x/2)}{\sin^2(\delta/2)}\right) \geq \tan\left(\frac{\delta}{2}\right)\left(1 - \frac{x^2}{4\sin^2(\delta/2)}\right).\end{aligned}$$

Substituting this into (3.27) we have

$$T_d\left(1 + 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}\right) \geq \frac{1}{2}e^{\sqrt{2}d\tan(\delta/2)}\exp\left(-\frac{dx^2}{\sqrt{2}\sin(\delta)}\right).$$

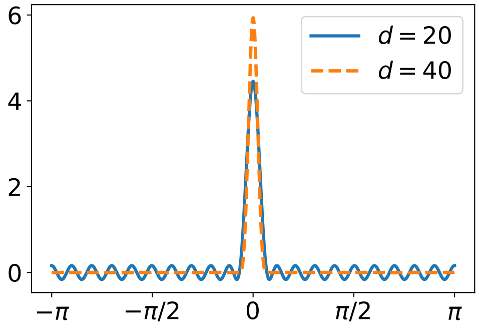


Figure 3.5: Illustration of $M_{d,\delta}(x)$ for $\delta = 0.2$, $d = 20, 40$.

Then

$$\begin{aligned}\mathcal{N}_{d,\delta} &\geq \int_{-\delta}^{\delta} T_d\left(1 + 2\frac{\cos(x) - \cos(\delta)}{1 + \cos(\delta)}\right) \mathrm{d}x - 2\pi \\ &\geq \frac{1}{2} e^{\sqrt{2}d\tan(\delta/2)} \sqrt{\frac{\sqrt{2}\pi \sin(\delta)}{d}} \operatorname{erf}\left(\sqrt{\frac{d}{\sqrt{2}\sin(\delta)}}\delta\right) - 2\pi \\ &\geq C_1 e^{d\delta/\sqrt{2}} \sqrt{\frac{\delta}{d}} \operatorname{erf}(C_2\sqrt{d\delta}),\end{aligned}$$

for $\delta \in (0, \pi/2)$ and some constants $C_1, C_2 > 0$. This proves (iii).

□

A plot of $M_{d,\delta}$ is shown in Figure 3.5. As we can see it roughly takes the shape of a Dirac function. We then use it as a mollifier to approximate the Heaviside function using the convolution of $M_{d,\delta}$ and the Heaviside function.

Lemma 18. *Let $H(x)$ be the periodic Heaviside function defined in* (3.5)*. For any $\delta \in (0, \pi/2)$ such that $\tan(\delta/2) \leq 1 - 1/\sqrt{2}$ and $\epsilon > 0$, there exists $d = \mathcal{O}(\delta^{-1}\log(\delta^{-1}\epsilon^{-1}))$, and a 2π-periodic function $F_{d,\delta}(x)$ of the form*

$$F_{d,\delta}(x) = \frac{1}{\sqrt{2\pi}} \sum_{k=-d}^{d} \hat{F}_{d,\delta,k} e^{ikx},$$

satisfying

(i) $-\epsilon/2 \leq F_{d,\delta}(x) \leq 1 + \epsilon$ for all $x \in \mathbb{R}$;

(ii) $|F_{d,\delta}(x) - H(x)| \leq \epsilon$ *for all* $x \in [-\pi + \delta, -\delta] \cup [\delta, \pi - \delta]$*;*

(iii) $|\hat{F}_{d,\delta,k}| \leq 2(1+\epsilon)/(\sqrt{2\pi}|k|)$ *for* $k \neq 0$*.*

Proof. We first construct the function $F_{d,\delta}(x)$. Let $M_{d,\delta}(x)$ be the mollifier in Lemma 17. Because of Lemma 17 (i) and (ii) $M_{d,\delta}(x)$ can be used as to mollify non-smooth functions. Also because $T_d(x)$ is a polynomial of degree d, the Fourier coefficients

$$\hat{M}_{d,\delta,k} = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} M_{d,\delta}(x) e^{-ikx} \mathrm{d}x$$

are non-zero only for $-d \leq k \leq d$. Also

$$\left|\hat{M}_{d,\delta,k}\right| \leq \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} |M_{d,\delta}(x)| \mathrm{d}x = \frac{1+\epsilon}{\sqrt{2\pi}}. \tag{3.28}$$

We construct $F_{d,\delta}$ by mollifying the Heaviside function with $M_{d,\delta}(x)$:

$$F_{d,\delta}(x) = (M_{d,\delta} * H)(x) = \int_{-\pi}^{\pi} M_{d,\delta}(x') H(x - x') \mathrm{d}x'. \tag{3.29}$$

We then show we can choose $d = \mathcal{O}(\delta^{-1}\log(\delta^{-1}\epsilon^{-1}))$ to satisfy (ii). We have

$$\begin{aligned} |F_{d,\delta}(x) - H(x)| &= \left| \int_{-\pi}^{\pi} M_{d,\delta}(x')(H(x - x') - H(x)) \mathrm{d}x' \right| \\ &\leq \int_{-\pi}^{\pi} M_{d,\delta}(x') |H(x - x') - H(x)| \mathrm{d}x'. \end{aligned}$$

For any x such that $|x| \in [\delta, \pi - \delta]$, first we consider the case where $|x'| < \delta$. In this case $H(x - x') = H(x)$ and therefore the integrand $M_{d,\delta}(x')|H(x - x') - H(x)| = 0$. Then we consider the case where $|x'| \geq \delta$. By Lemma 17 (i) we have $M_{d,\delta}(x') \leq 2/\mathcal{N}_{d,\delta}$, and as $|H(x - x') - H(x)| \leq 1$, $M_{d,\delta}(x')|H(x - x') - H(x)| \leq 2/\mathcal{N}_{d,\delta}$. Thus for any x such that $|x| \in [\delta, \pi - \delta]$,

$$|F_{d,\delta}(x) - H(x)| \leq \frac{4\pi}{\mathcal{N}_{d,\delta}}. \tag{3.30}$$

If we want to keep the approximation error for $x \in [-\pi + \delta, -\delta] \cup [\delta, \pi - \delta]$ to be below ϵ, we will need, by Lemma 17 (i) and (3.30),

$$C_1 e^{d\delta/\sqrt{2}} \sqrt{\frac{\delta}{d}} \operatorname{erf}(C_2\sqrt{d\delta}) \geq \frac{4\pi}{\epsilon}.$$

It can be checked that we can choose $d = \mathcal{O}(\delta^{-1}\log(\epsilon^{-1}\delta^{-1}))$ to achieve this.

We then show this choice of d ensures (i) as well. From eq. (3.26)

$$F_{d,\delta}(x) \leq \int_{-\pi}^{\pi} |M_{d,\delta}(y)|\mathrm{d}y \leq 1 + \frac{4\pi}{\mathcal{N}_{d,\delta}} \leq 1 + \epsilon$$

and by the second inequality in Lemma 17 (i)

$$F_{d,\delta}(x) \geq -\frac{1}{\mathcal{N}_{d,\delta}} \int_{-\pi}^{\pi} H(y)\mathrm{d}y = -\frac{2\pi}{\mathcal{N}_{d,\delta}} \geq -\frac{\epsilon}{2}.$$

Finally we prove our construction satisfies (iii). Because $F_{d,\delta}(x)$ is defined through a convolution, its Fourier coefficients can be obtained through

$$\hat{F}_{d,\delta,k} = \sqrt{2\pi}\hat{M}_{d,\delta,k}\hat{H}_k,$$

where $\hat{H}_k$'s are the Fourier coefficients of the rectangle function $H(x)$. Therefore $\hat{F}_{d,\delta,k} \neq 0$ only for $-d \leq k \leq d$. Because of (3.28), we have

$$|\hat{F}_{d,\delta,k}| \leq (1+\epsilon)|\hat{H}_k|.$$

Since when $k \neq 0$

$$\begin{aligned}\hat{H}_k &= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} H(x)e^{-ikx}\mathrm{d}x \\ &= \begin{cases} \frac{2}{i\sqrt{2\pi}k} & 2 \nmid k \\ 0 & 2 \mid k \end{cases}\end{aligned}$$

we have (iii).

□

3.8 The relation between the CDF and the approximate CDF

In this appendix we prove (3.9). Let $0 < \delta < \pi/6$. First we have a 2π-periodic function $F(x)$ from Lemma 18 that satisfies

$$|F(x) - H(x)| \leq \epsilon, \quad x \in [-\pi+\delta, -\delta] \cup [\delta, \pi-\delta],$$

and $F(x) \in [0,1]$ for all $x \in \mathbb{R}$. We further define $F_L(x) = F(x-\delta)$ and $F_R(x) = F(x+\delta)$. They satisfy

$$\begin{aligned} |F_L(x) - H(x)| &\leq \epsilon, \quad x \in [-\pi+2\delta, 0] \cup [2\delta, \pi], \\ |F_R(x) - H(x)| &\leq \epsilon, \quad x \in [-\pi, -2\delta] \cup [0, \pi-2\delta]. \end{aligned} \tag{3.31}$$

We define the some functions related to the ACDF as follows:

$$\widetilde{C}_L(x) = (F_L * p)(x), \quad \widetilde{C}_R(x) = (F_R * p)(x). \tag{3.32}$$

Then we have

$$\widetilde{C}_L(x) = \widetilde{C}(x-\delta), \quad \widetilde{C}_R(x) = \widetilde{C}(x+\delta). \tag{3.33}$$

The functions $\widetilde{C}_L(x)$ and $\widetilde{C}_R(x)$ can be used to bound $C(x)$. Because of (3.31), the fact that $p(x)$ is supported in $(-\pi/3, \pi/3)$ in $[-\pi, \pi]$, $\delta < \pi/6$, and that $H(y)$ and $F_L(y)$ both take value in $[0, 1]$, for $x \in (-\pi/3, \pi/3)$ we have

$$\begin{aligned} |\widetilde{C}_L(x) - C(x)| &\leq \int_{-\pi}^{\pi} p(x-y)|H(y) - F_L(y)|\mathrm{d}y \\ &\leq \epsilon + \int_0^{2\delta} p(x-y)\mathrm{d}y \\ &= \epsilon + C(x) - C(x-2\delta). \end{aligned}$$

Therefore

$$\widetilde{C}_L(x) \geq C(x) - [\epsilon + C(x) - C(x-2\delta)] = C(x-2\delta) - \epsilon.$$

Similarly we have

$$\widetilde{C}_R(x) \leq C(x) + [\epsilon + C(x+2\delta) - C(x)] = C(x+2\delta) + \epsilon.$$

Combining these two inequalities with (3.33), we have

$$C(x-2\delta) \leq \widetilde{C}(x-\delta) + \epsilon, \quad C(x+2\delta) \geq \widetilde{C}(x+\delta) - \epsilon.$$

This proves (3.9).

3.9 Obtaining the ground state energy by solving the QEEP

Here we discuss how to obtain the ground state energy using algorithm in Ref. [149] to solve the QEEP. The cost of solving the QEEP as analyzed in Ref. [149] scales as ϵ^{-6}. However, the cost can be much reduced for the problem of ground state energy estimation. For simplicity we assume $\|H\| < \pi/3$ and τ is chosen to be 1.

In order to find the interval of size 2ϵ containing the ground state energy , we first divide the interval $[-\pi/3, \pi/3]$ into M bins of equal size smaller than 2ϵ. We then define the indicator function associated with an interval $[a, b]$ to be

$$1_{[a,b]}(x) = \begin{cases} 1, & x \in [a,b], \\ 0, & x \notin [a,b]. \end{cases}$$

In QEEP the goal is to estimate $\mathrm{Tr}[\rho 1_{[a,b]}(H)]$, where $[a, b]$ is one of the M bins, to within precision $\mathcal{O}(\epsilon)$. However, in our setting, if we know $p_0 \geq \eta$, one can estimate $\mathrm{Tr}[\rho 1_{[a,b]}(H)]$ to within error $\mathcal{O}(\eta)$. If we get $\mathrm{Tr}[\rho 1_{[a,b]}(H)] < \eta$ with high confidence then we know the ground state energy λ_0 is not in this interval. If know $\mathrm{Tr}[\rho 1_{[a,b]}(H)] > \eta/2$ with high confidence then there is an eigenvalue in $[a, b]$. If the above task can be done, then we choose the leftmost bin in which $\mathrm{Tr}[\rho 1_{[a,b]}(H)] > \eta/2$. This will enable us to solve the ground state energy estimation problem.

To estimate $\mathrm{Tr}[\rho 1_{[a,b]}(H)]$, Ref. [149] first approximated the indicator function $1_{[a,b]}(x)$ using a truncated Fourier series [149, Appendix A], similar to what we did in section 3.7. The number of terms N_{term} and the maximal evolution time T both scale like ϵ^{-1}. In Ref. [149] the author proposed estimating each Fourier mode $\mathrm{Tr}[\rho e^{-ijH}]$ to within error $\mathcal{O}(\epsilon/N_{\mathrm{term}})$. Because here the estimation precision is $\mathcal{O}(\eta)$ rather than $\mathcal{O}(\epsilon)$, we should instead estimate $\mathrm{Tr}[\rho e^{-ijH}]$ to within error $\mathcal{O}(\eta/N_{\mathrm{term}}) = \mathcal{O}(\eta\epsilon)$. Because we are using Monte Carlo sampling this requires $\mathcal{O}(\eta^{-2}\epsilon^{-2})$ samples. We need the same number of samples for each $\mathrm{Tr}[\rho e^{-ijH}]$, and therefore the total time we need to run time evolution is $\mathcal{O}(N_{\mathrm{term}} T \eta^{-2}\epsilon^{-2}) = \mathcal{O}(\eta^{-2}\epsilon^{-4})$. We omitted polylogarithmic factors in the complexity.

However if the analysis is done more carefully the dependence on ϵ could be improved. First one should notice that the error for each $\mathrm{Tr}[\rho e^{-ijH}]$ is independent, and the estimate is unbiased (if we do not consider the Fourier approximation error), as is the case in our algorithm (Section 3.3). Therefore the total error for estimating $\mathrm{Tr}[\rho 1_{[a,b]}(H)]$ accumulates sublinearly. More precisely, let the error for estimating $\mathrm{Tr}[\rho e^{-ijH}]$ be ε_j with variance σ_j^2, and let the coefficient for $\mathrm{Tr}[\rho e^{-ijH}]$ be A_j. Then the total error $\sum_j A_j \varepsilon_j$ has variance $\sum_j A_j^2 \sigma_j^2$. Therefore the total error is roughly $\sqrt{\sum_j A_j^2 \sigma_j^2}$ instead of the linearly accumulated error $\sum_j A_j \sigma_j$. These two can have different asymptotic scaling depending on the magnitude of A_j. Because of this one can in fact choose to estimate $\mathrm{Tr}[\rho e^{-ijH}]$ to within error $\mathcal{O}(\eta/\sqrt{N_{\mathrm{term}}}) = \mathcal{O}(\eta\epsilon^{-1/2})$. This saves a ϵ^{-1} factor in the total runtime. Furthermore, one can choose to evaluate the approximate indicator function in a stochastic way, like we did in Section 3.3. By taking into account the decay of Fourier coefficients, similar to Lemma 18 (iii), it is possible to further reduce the complexity.

3.10 Complexity analysis for using Trotter formulas

In this appendix, instead of using the maximal evolution time and the total evolution time to quantify the complexity, we directly analyze the circuit depth and the total runtime when the time evolution is simulated using Trotter formulas. We suppose the Hamiltonian H can be decomposed as $H = \sum_\gamma H_\gamma$, where each of H_γ can be efficiently exponentiated. A p-th order Trotter formula applied to $e^{-i\tau H}$ with r Trotter steps gives us a unitary operator U_{HS} with error

$$\|U_{\mathrm{HS}} - e^{-i\tau H}\| \leq C_{\mathrm{Trotter}} \tau^{p+1} r^{-p},$$

where C_{Trotter} is a prefactor, for which the simplest bound is $C_{\mathrm{Trotter}} = \mathcal{O}((\sum_\gamma \|H\|_\gamma)^{p+1})$. Tighter bounds in the form of a sum of commutators are proved in Refs. [60, 152].

The algorithm in this work

Our algorithm requires approximating Eq. (3.10) to precision η (as in Theorem 15 η is a lower bound of $p_0/2$) using Trotter formulas. Suppose we are using a p-th order Trotter formula, then we want

$$\left\| \sum_j \hat{F}_j e^{ijx} \mathrm{Tr}[\rho e^{-ij\tau H}] - \sum_j \hat{F}_j e^{ijx} \mathrm{Tr}[\rho U_{\mathrm{HS}}^j] \right\| = \mathcal{O}(\eta).$$

Since the left-hand side can be upper bounded by

$$\sum_j |\hat{F}_j| |j| \|e^{-i\tau H} - U_{\mathrm{HS}}\| = \mathcal{O}(d \|e^{-i\tau H} - U_{\mathrm{HS}}\|)$$

by Lemma 18 (iii), we only need to choose r so that

$$C_{\mathrm{Trotter}} \tau^{p+1} r^{-p} = \mathcal{O}(\eta d^{-1}).$$

Therefore we can choose

$$r = \max\{1, \widetilde{\mathcal{O}}(d^{1/p} \eta^{-1/p} C_{\mathrm{Trotter}}^{1/p} \tau^{1+1/p})\}$$

The maximal evolution time in Corollary 15 tells us how many times we need to use the operator U_{HS} (multiplied by a factor τ). Multiply this by r we have the maximal circuit depth we need, which is

$$dr = \widetilde{\mathcal{O}}(\max\{\tau^{-1}\epsilon^{-1}, \epsilon^{-1-1/p} \eta^{-1/p} C_{\mathrm{Trotter}}^{1/p}\}). \tag{3.34}$$

Similarly we have the total runtime

$$\widetilde{\mathcal{O}}(\max\{\tau^{-1}\epsilon^{-1}\eta^{-2}, \epsilon^{-1-1/p}\eta^{-2-1/p}C_{\text{Trotter}}^{1/p}\}). \tag{3.35}$$

If we fix H and let $\epsilon, \eta \to 0$, then we can see this gives us an extra $\epsilon^{-1/p}\eta^{-1/p}$ factor in the circuit depth and total runtime, compared to the maximal evolution time and the total evolution time respectively.

Quantum phase estimation

We then analyze the circuit depth and total runtime requirement for estimating the ground state energy with QPE, where the time evolution is performed using Trotter formulas. We analyze the multi-ancilla qubit version of QPE and the result is equally valid for the single-ancilla qubit version using semi-classical Fourier transform.

In QPE, when we replace all exact time evolution with U_{HS}, we would like to ensure that the probability of obtaining an energy measurement close to the ground state energy remains bounded away from 0 by $\Omega(\eta)$. Therefore the probability distribution of the final measurement outcome should be at most $\mathcal{O}(\eta)$ away from the original distribution in terms of the total variation distance.

Because the only part of QPE that depends on the time evolution operator is the multiply-controlled unitary

$$\sum_{j=0}^{J-1} |j\rangle\langle j| \otimes e^{-ij\tau H},$$

which is replaced by

$$\sum_{j=0}^{J-1} |j\rangle\langle j| \otimes U_{\text{HS}}^{j}$$

when we use Trotter formulas, we only need to ensure the difference between the two operators to be upper bounded by $\mathcal{O}(\eta)$ in terms of operator norm. Therefore we need

$$J\|e^{-ij\tau H} - U_{\text{HS}}\| = \mathcal{O}(\eta).$$

As discussed in Section 3.1, we need to choose $J = \mathcal{O}(\tau^{-1}\epsilon^{-1}\eta^{-1})$ (we need the τ^{-1} factor to account for rescaling H, and p_0 in Section 3.1 is replaced by η). Following the same analysis as in the previous section, we need to choose the number of Trotter steps for approximating $e^{-i\tau H}$ to be

$$r = \max\{1, \mathcal{O}(J^{1/p}\eta^{-1/p}C_{\text{Trotter}}^{1/p}\tau^{1+1/p})\}$$

Therefore the circuit depth needed is

$$Jr = \mathcal{O}(\max\{\tau^{-1}\epsilon^{-1}\eta^{-1}, \epsilon^{-1-1/p}\eta^{-1-2/p}C_{\text{Trotter}}^{1/p}\}), \tag{3.36}$$

and the total runtime is

$$\mathcal{O}(\max\{\tau^{-1}\epsilon^{-1}\eta^{-2}, \epsilon^{-1-1/p}\eta^{-2-2/p}C_{\text{Trotter}}^{1/p}\}). \tag{3.37}$$

Again, if we fix H and let $\epsilon, \eta \to 0$, then we can see this gives us an extra $\epsilon^{-1/p}\eta^{-2/p}$ factor in the circuit depth and total runtime, compared to the maximal evolution time and the total evolution time respectively. This is worse by a factor of $\eta^{-1/p}$ than the cost using our algorithm.

3.11 The control-free setting

In this appendix we introduce, as an alternative to the quantum circuit in (3.1), a circuit which does not require controlled time evolution. This construction is mainly based on the ideas in Refs. [142, 118, 127]. We will introduce the construction of the circuit and discuss how to use the measurement results from the circuit to construct a random variable $\widetilde{Z}$ satisfying

$$\mathbb{E}[\widetilde{Z}] = \mathrm{Tr}[\rho e^{-itH}] \tag{3.38}$$

for any given t. Then choosing $t = j\tau$, we will be able to replace X_j and Y_j with $\mathrm{Re}\,\widetilde{Z}$ and $\mathrm{Im}\,\widetilde{Z}$ respectively, while satisfying (3.2) and (3.3). In order to remove the need of performing controlled time evolution of H, we need some additional assumptions.

1. The initial state ρ is a pure state $|\phi_0\rangle$, prepared using a unitary circuit U_I.
2. We have a reference eigenstate $|\psi_R\rangle$ of H corresponding to a known eigenvalue λ_R. This eigenstate can be efficiently prepared using a unitary circuit U_R.
3. $\langle\psi_R|\phi_0\rangle = 0$.

The last assumption $\langle\psi_R|\phi_0\rangle = 0$ implies $\langle\psi_R|e^{-itH}|\phi_0\rangle = 0$ for all $t \in \mathbb{R}$ because $|\psi_R\rangle$ is an eigenvector of e^{-itH}. All of these are reasonable assumptions for a second-quantized fermionic Hamiltonian: we choose $|\psi_R\rangle$ to be the vacuum state, $\lambda_R = 0$, and $|\phi_0\rangle$ to be the Hartree-Fock state, which can be efficiently prepared [104]. Naturally $\langle\psi_R|\phi_0\rangle = 0$ because of the particle number conservation.

With these assumptions, we let

$$\alpha = \langle\phi_0|e^{-it(H-\lambda_R)}|\phi_0\rangle\,.$$

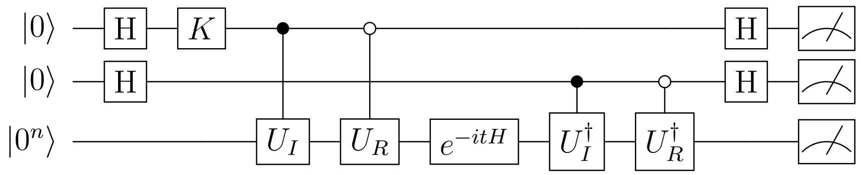
Also define

$$|\Psi_{0,\pm}\rangle = \frac{1}{\sqrt{2}}(|\psi_R\rangle \pm |\phi_0\rangle), \quad |\Psi_{1,\pm}\rangle = \frac{1}{\sqrt{2}}(|\psi_R\rangle \pm i\,|\phi_0\rangle).$$

With these states, we can express α in terms of expectation values:

$$\begin{aligned}\langle\Psi_{0,+}|e^{-itH}|\Psi_{0,\pm}\rangle &= \frac{1}{2}e^{-i\lambda_R t}(1\pm\alpha),\\ \langle\Psi_{0,+}|e^{-itH}|\Psi_{1,\pm}\rangle &= \frac{1}{2}e^{-i\lambda_R t}(1\pm i\alpha).\end{aligned}$$

In Refs. [118, 142] it is assumed that we have unitary circuits to prepare $|\Psi_{0,\pm}\rangle$ and $|\Psi_{1,\pm}\rangle$. However it is not immediately clear how these circuits are constructed. Here we will take a slightly different approach. The circuit diagram is as follows:



(3.39)

In this circuit we choose $K = I$ for the real part of α or the phase gate S for the imaginary part of α. This circuit uses three registers, with the first two containing one qubit each, and the third one containing n qubits.

We first analyze the probability of different measurement outcomes when $K = I$. When we run the above circuit, and measure all the qubits, the probability of the measurement outcomes of the first two qubits being (b_1, b_2), and the rest of the qubits being all 0, is

$$\begin{aligned}p_{0,(b_1,b_2)} &= \begin{cases} |\langle\Psi_{0,+}|e^{-itH}|\Psi_{0,+}\rangle|^2/4, & b_1 = b_2\\ |\langle\Psi_{0,+}|e^{-itH}|\Psi_{0,-}\rangle|^2/4, & b_1 \neq b_2\end{cases}\\ &= \frac{1}{16}(1+|\alpha|^2+2(-1)^{b_1+b_2}\operatorname{Re}\alpha).\end{aligned}$$

Here we have used the fact that $|\langle\Psi_{0,+}|e^{-itH}|\Psi_{0,+}\rangle| = |\langle\Psi_{0,-}|e^{-itH}|\Psi_{0,-}\rangle|$.

Similarly, when $K = S$, the probability of the measurement outcomes of the first two qubits being (b_1, b_2), and the rest of the qubits being all 0, is

$$\begin{aligned}p_{1,(b_1,b_2)} &= \begin{cases} |\langle\Psi_{0,+}|e^{-itH}|\Psi_{1,+}\rangle|^2/4, & b_1 = b_2\\ |\langle\Psi_{0,+}|e^{-itH}|\Psi_{1,-}\rangle|^2/4, & b_1 \neq b_2\end{cases}\\ &= \frac{1}{16}(1+|\alpha|^2-2(-1)^{b_1+b_2}\operatorname{Im}\alpha).\end{aligned}$$

Based on the above analysis, we construct the random variable $\widetilde{Z}$ in the following way: we first run the circuit with $K = I$, and denote the measurement outcomes of the first two qubits by (b_1, b_2). If the third register returns all 0 when measured, then we let $\widetilde{X} = (-1)^{b_1+b_2}$. Otherwise we let $\widetilde{X} = 0$. Similarly we define a random variable $\widetilde{Y}$ for $K = S$. We have

$$\mathbb{E}[\widetilde{X}] = p_{0,(0,0)} + p_{0,(1,1)} - p_{0,(0,1)} - p_{0,(1,0)} = \frac{1}{2} \operatorname{Re} \alpha,$$

and

$$\mathbb{E}[\widetilde{Y}] = p_{1,(0,0)} + p_{1,(1,1)} - p_{1,(0,1)} - p_{1,(1,0)} = -\frac{1}{2} \operatorname{Im} \alpha.$$

Therefore we can define

$$\widetilde{Z} = 2e^{-i\lambda_R t}(\widetilde{X} - i\widetilde{Y}).$$

Then

$$\mathbb{E}[\widetilde{Z}] = e^{-i\lambda_R t}\alpha = \operatorname{Tr}[\rho e^{-itH}].$$

Thus we can see this new random variable $\widetilde{Z}$ satisfies (3.38). Compared to the Z in the main text this new random variable has a slightly larger variance:

$$\operatorname{var}[\widetilde{Z}] \leq \mathbb{E}[|\widetilde{Z}|^2] \leq 8.$$

This however does not change the asymptotic complexity.

3.12 Details on the numerical experiments

In Figure 3.3, we apply the procedure described in Section 3.3 to approximate the CDF of the Fermi-Hubbard model, described by the Hamiltonian

$$H = -t \sum_{\langle j,j' \rangle,\sigma} c_{j,\sigma}^\dagger c_{j',\sigma} + U \sum_j \left(n_{j,\uparrow} - \frac{1}{2}\right)\left(n_{j,\downarrow} - \frac{1}{2}\right), \tag{3.40}$$

where $c_{j,\sigma}$ ($c_{j,\sigma}^\dagger$) denotes the fermionic annihilation (creation) operator on the site j with spin $\sigma \in \{\uparrow, \downarrow\}$. $\langle \cdot, \cdot \rangle$ denotes sites that are adjacent to each other. $n_{j,\sigma} = c_{j,\sigma}^\dagger c_{j,\sigma}$ is the number operator. The sites are arranged into a one-dimensional chain, with open boundary condition.

We first evaluate $\bar{G}(x)$ defined in (3.16), and the result is shown in Figure 3.3. We use a classical computer to simulate the sampling from the quantum circuit. The initial state ρ is chosen to be the Hartree-Fock solution, which has an overlap of around 0.4 with the exact ground state. We can see that $\bar{G}(x)$ closely follows the

CDF, and even though there is significant noise from Monte Carlo sampling, the jump corresponding to the ground state energy is clearly resolved.

Then we consider estimating the ground state energy from $\bar{G}(x)$. In this numerical experiment we use a heuristic approach, and the rigorous approach that comes with provable error bound and confidence level is discussed in Sections 3.4 and 3.5. We obtain the estimate by

$$x^\star = \inf\{x : \bar{G}(x) \geq \eta/2\},$$

and $x^\star/\tau$ is an estimate for the ground state energy λ_0. We expect $x^\star \in [\tau\lambda_0 - \delta, \tau\lambda_0 + \delta]$. Here η is chosen so that $p_0 \geq \eta$.

The error of the estimated ground state energy, the total evolution time, and the maximal evolution time are shown in Figure 3.4, in which we have chosen $U/t = 4$ for the Hubbard model. In the right panel of Figure 3.4 we can see the line for total evolution time runs parallel to the line for the maximal evolution time. Because the maximal evolution time scales linearly with respect to δ^{-1}, and this plot uses logarithmic scales for both axes, we can see the total evolution time has a δ^{-1} scaling, and is therefore inversely proportional to the allowed error of ground state energy estimation.

3.13 Frequently used symbols

Symbol	Meaning
H	The Hamiltonian for which we want to estimate the ground state energy.
ρ	The initial state from which we perform time evolution and measurement.
p_k	The overlap between ρ and the k-th lowest eigensubspace.
τ	A renormalization factor satisfying $\tau\|H\| \leq \pi/4$.
$p(x)$	The spectral density associated with τH and ρ.
$C(x)$	The cumulative distribution function defined in (3.6).
$\widetilde{C}(x)$	The approximate CDF defined in (3.8).
$G(x)$	An unbiased estimate of the ACDF $\widetilde{C}(x)$ defined in (3.14).
$\bar{G}(x)$	The average of multiple samples of $G(x)$, defined in (3.16).
J_k	An integer drawn from the distribution (3.11) signifying the number of steps in the time evolution. $\lvert J_k\rvert \leq d$.
Z_k	A sample generated on a quantum circuit from two measurement outcomes. Defined in (3.13). Can only take value $\pm 1 \pm i$.
d	The maximal possible value of $\lvert J_k\rvert$.
δ	In the context of Corollary 15 we choose $\delta = \tau\epsilon$ where ϵ is the allowed error of the ground state energy.
ϑ	The allowed failure probability.

Table 3.3: Frequently used symbols in this chapter.

Chapter 4

Solving quantum linear systems

We present a quantum eigenstate filtering algorithm based on quantum signal processing (QSP) and minimax polynomials. The algorithm allows us to efficiently prepare a target eigenstate of a given Hamiltonian, if we have access to an initial state with non-trivial overlap with the target eigenstate and have a reasonable lower bound for the spectral gap. We apply this algorithm to the quantum linear system problem (QLSP), and present two algorithms based on quantum adiabatic computing (AQC) and quantum Zeno effect respectively. Both algorithms prepare the final solution as a pure state, and achieves the near optimal $\widetilde{\mathcal{O}}(d\kappa\log(1/\epsilon))$ query complexity for a d-sparse matrix, where κ is the condition number, and ϵ is the desired precision. Neither algorithm uses phase estimation or amplitude amplification.

4.1 Introduction

Eigenvalue problems have a wide range of applications in scientific and engineering computing. Finding ground states and excited states of quantum many-body Hamiltonian operators, Google's PageRank algorithm, and principle component analysis are just a few prominent examples. Some problems that are not apparently eigenvalue problems may benefit from a reformulation into eigenvalue problems. One noticeable example is the quantum linear systems problem (QLSP), which aims at preparing a state that is proportional to the solution of a given linear system, i.e. $|x\rangle = A^{-1}|b\rangle/\|A^{-1}|b\rangle\|$ on a quantum computer ($\|\cdot\|$ denotes the vector 2-norm). Here $A \in \mathbb{C}^{N\times N}$, and $|b\rangle \in \mathbb{C}^N$. We give a more detailed definition of the QLSP in Section 4.4. All QLSP solvers share the desirable property that the complexity with respect to the matrix dimension can be as low as $\mathcal{O}(\text{polylog}(N))$, which is exponentially faster compared to known classical solvers. Due to the wide applications of linear systems, the efficient solution of QLSP has received significant attention in recent years [93, 59, 86, 154, 15, 49, 172, 48, 176, 43]. By reformulating QLSP into an eigenvalue problem, recent developments have yielded near-optimal query-complexity with respect to κ (the condition number of A, defined as the ratio between the largest and the smallest singular value of A, or $\kappa = \|A\|\|A^{-1}\|$) [154, 15], which is so far difficult to achieve using alternative methods.

Consider a Hermitian matrix $H \in \mathbb{C}^{N\times N}$, which has a known interior eigenvalue λ separated from the rest of the spectrum by a gap (or a lower bound of the gap) denoted by Δ. Let P_λ be the spectral projector associated with the eigenvalue λ. The goal of the quantum eigenstate filtering problem is to find a certain smooth function $f(\cdot)$, so that $\|f(H-\lambda I) - P_\lambda\|$ is as small as possible, and there should be a unitary quantum circuit U that efficiently implements $f(H-\lambda I)$. Then given an initial state $|x_0\rangle$ so that $\|P_\lambda |x_0\rangle\| = \gamma > 0$, $f(H-\lambda I)|x_0\rangle$ filters out the unwanted spectral components in $|x_0\rangle$ and is approximately an eigenstate of H corresponding to λ. We assume that H can be block-encoded into a unitary matrix U_H [83], which is our input model for H and requires a certain amount of ancilla qubits. The initial state is prepared by an oracle U_{x_0}. In this paper when comparing the number of qubits needed, we focus on the *extra* ancilla qubits introduced by the various methods used, which exclude the ancilla qubits used in the block-encoding of H.

In this paper, we develop a polynomial-based filtering method, which chooses $f = P_\ell$ to be a ℓ-th degree polynomial. We prove that our choice yields the *optimal* compression ratio among all polynomials. Assume that the information of H can be accessed through its block-encoding. Then we demonstrate that the optimal eigenstate filtering polynomial can be efficiently implemented using the recently developed quantum signal processing (QSP) [86, 115], which allows us to implement a general matrix polynomial with a minimal number of ancilla qubits. More specifically, the query complexity of our method is $\widetilde{\mathcal{O}}(1/(\gamma\Delta)\log(1/\epsilon))$ for the block-encoding of the Hamiltonian and $\mathcal{O}(1/\gamma)$ for initial state preparation, when using amplitude amplification. The number of extra ancilla qubits is merely 3 when using amplitude amplification, and 2 when we do not (in this case the $1/\gamma$ factor in both query complexities become $1/\gamma^2$). However in the application to QLSP we can always guarantee $\gamma = \Omega(1)$, and thus not using amplitude amplification only changes the complexity by a constant factor.

Using the quantum eigenstate filtering algorithm, we present two algorithms to solve QLSP, both achieving a query complexity $\widetilde{\mathcal{O}}(\kappa\log(1/\epsilon))$, with constant success probability (success is indicated by the outcome of measuring the ancilla qubits). For any $\delta > 0$, a quantum algorithm that is able to solve a generic QLSP with cost $\mathcal{O}(\kappa^{1-\delta})$ would imply $\mathsf{BQP}{=}\mathsf{PSPACE}$ [93]. Therefore our algorithm is near-optimal with respect to κ up to a logarithmic factor, and is optimal with respect to ϵ. The first algorithm (Theorem 26) combines quantum eigenstate filtering with the time-optimal adiabatic quantum computing (AQC) approach [15]. We use the time-optimal AQC to prepare an initial state $|x_0\rangle$, which achieves a nontrivial overlap with the true solution as $\gamma = |\langle x_0|x\rangle| \sim \Omega(1)$. Then we apply the eigenstate filtering to $|x_0\rangle$ once, and the filtered state is ϵ-close to $|x\rangle$ upon measurement. The second algorithm (Theorem 29) combines quantum eigenstate filtering with the time-optimal version of the approach based on the quantum Zeno effect (QZE) [39, 154]. Instead of preparing one initial vector satisfying $\gamma \sim \Omega(1)$, a sequence of quantum eigenstate filtering algorithm are applied to obtain to the instantaneous eigenstate of interest along an eigenpath. The final state is again ϵ-close to $|x\rangle$ upon measurement. Neither algorithm involves phase estimation or any form of amplitude amplification. The first algorithm achieves slightly better dependence on κ than the second algorithm, but this comes at the expense of using a time-dependent Hamiltonian simulation procedure [116] resulting in this algorithm using more ancilla qubits than the second QZE-based algorithm. For both algorithms, because the success is indicated by the outcome of measuring the ancilla qubits, we can repeat the algorithms $\mathcal{O}(\log(1/\delta))$ times to boost the final success probability from $\Omega(1)$ to $1-\delta$ for arbitrarily small δ.

Related works:

A well-known quantum eigenstate filtering algorithm is phase estimation [101], which relies on Hamiltonian simulation [113, 35, 116, 115, 114, 34] and the quantum Fourier transform. We treat the Hamiltonian simulation $e^{-\mathrm{i}H\tau}$ with some fixed τ as an oracle called U_{sim}, where τ satisfies $\tau\|H\| < \pi$. Ref. [80, Appendix B] contains a very detailed analysis of the complexities of using phase estimation together with amplitude amplification. From the analysis in Ref. [80], this approach requires $\widetilde{\mathcal{O}}(1/(\gamma^2\Delta\epsilon))$ times of queries for U_{sim}, where ϵ is the target accuracy (the complexity is the same up to logarithmic factors if we use the block-encoding U_H instead of its time-evolution as an oracle); the number of queries to the circuit U_{x_0} that prepares the initial trial state is $\widetilde{\mathcal{O}}(1/\gamma)$; and the number of extra ancilla qubits is $\mathcal{O}(\log(1/(\epsilon\Delta))$. This is non-optimal with respect to both γ and ϵ.

Several variants of phase estimation are developed to achieve better dependence on the parameters γ and ϵ [134, 135, 80]. The filtering method developed by Poulin and Wocjan [134] (for a task related to eigenstate filtering) improves the query complexities of U_{sim} and U_{x_0} with respect to γ from $\widetilde{\mathcal{O}}(1/\gamma^2)$ to $\widetilde{\mathcal{O}}(1/\gamma)$. Ge et al. [80, Appendix C] shows that the method by Poulin and Wocjan can be adapted to the ground state preparation problem so that the query complexity of U_{sim} becomes $\widetilde{\mathcal{O}}(1/(\gamma\Delta)\log(1/\epsilon))$, while the complexity of U_{x_0} remains $\widetilde{\mathcal{O}}(1/\gamma)$. The number of extra ancilla qubits is $\mathcal{O}(\log(1/(\epsilon\Delta))$. Similar logarithmic dependence on the accuracy in the query complexity has also been achieved in Ref. [135].

Ge et al. [80] also proposed two eigenstate filtering algorithms using linear combination of unitaries (LCU) [59, 34], which uses the Fourier basis and the Chebyshev polynomial basis, respectively. For both methods, the query complexities for U_H and U_{x_0} are $\widetilde{\mathcal{O}}(1/(\gamma\Delta)\log(1/\epsilon))$ and $\widetilde{\mathcal{O}}(1/\gamma)$ respectively, and the number of extra ancilla qubits can be reduced to $\mathcal{O}(\log\log(1/\epsilon)+\log(1/\Delta))$. The $\log\log(1/\epsilon)$ factor comes from the use of LCU. We remark that these methods were developed for finding the ground state, but can be adapted to compute interior eigenstates as well. Our filtering method has the same query complexity up to polylogarithmic factors. The number of extra ancilla qubits is significantly fewer and does not depend on either ϵ or Δ, due to the use of QSP. Our method also uses the optimal filtering polynomial, which solves a minimax problem as recorded in Lemma 20. There are several other hybrid quantum-classical algorithms to compute ground state energy and to prepare the ground state [151, 132], whose computational complexities are not yet analyzed and therefore we do not make comparisons here.

For solving QLSP, the query complexity of the original Harrow, Hassidim, and Lloyd (HHL) algorithm [93] scales as $\widetilde{\mathcal{O}}(\kappa^2/\epsilon)$, where κ is the condition number of A, and ϵ is the target accuracy. Despite the exponential speedup with respect to the matrix dimension, the scaling with respect to κ and ϵ is significantly weaker compared to that in classical methods. For instance, for positive definite matrices, the complexity of steepest descent (SD) and conjugate gradient (CG) (with respect to both κ and ϵ) are only $\mathcal{O}(\kappa\log(1/\epsilon))$ and $\mathcal{O}(\sqrt{\kappa}\log(1/\epsilon))$, respectively [143].

In the past few years, there have been significant progresses towards reducing the pre-constants for quantum linear solvers. In particular, the linear combination of unitary (LCU) [34, 59] and quantum signal processing (QSP) or quantum singular value transformation (QSVT) [115, 86] techniques can reduce the query complexity to $\mathcal{O}(\kappa^2\operatorname{polylog}(\kappa/\epsilon))$. Therefore the algorithm is almost optimal with respect to ϵ, but is still suboptimal with respect to κ. The scaling with respect to κ can be reduced by the variable-time amplitude amplification (VTAA) [12] technique, and the resulting query complexity for solving QLSP is $\mathcal{O}(\kappa\operatorname{polylog}(\kappa/\epsilon)))$ [59, 49]. However, VTAA requires considerable modification of the LCU or QSP algorithm, and has significant overhead itself. To the extent of our knowledge, the performance of VTAA for solving QLSP has not been quantitatively reported in the literature.

Algorithm	Query complexity	Remark
HHL [93]	$\widetilde{\mathcal{O}}(\kappa^2/\epsilon)$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa/\epsilon^3)$ [11]
Linear combination of unitaries (LCU) [59, 49]	$\widetilde{\mathcal{O}}(\kappa^2 \operatorname{polylog}(1/\epsilon))$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$
Quantum singular value transformation (QSVT) [86]	$\widetilde{\mathcal{O}}(\kappa^2 \log(1/\epsilon))$	
Randomization method (RM) [154]	$\widetilde{\mathcal{O}}(\kappa/\epsilon)$	w. repeated phase estimation, complexity becomes $\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$
Time-optimal adiabatic quantum computing (AQC(exp)) [15]	$\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$	No need for any amplitude amplification. Use time-dependent Hamiltonian simulation.
Eigenstate filtering+AQC (Theorem 26)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification.
Eigenstate filtering+QZE (Theorem 29)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification. Does not rely on any complex subroutines.

Table 4.1: The number of queries to the block-encoding of the coefficient matrix A for solving QLSP. Some algorithms were not originally formulated using block-encoding as the input model, but can be converted to use the block-encoding model instead. In the HHL algorithm it is assumed that we have access to time-evolution under the Hermitian coefficient matrix as the Hamiltonian. This assumption can be met when we have the block-encoding of A using Hamiltonian simulation technique that results in small overhead [35, 116, 115, 114, 34]. The LCU method [59] and the gate-based implementation of the RM method [154] both assume oracles to access elements of A. However in both cases the oracles lead to a block-encoding A which can be used in the algorithms. The same can be said of the sparse-access input model in Ref. [49]. Time complexities and gate complexities are converted to query complexities with respect to the oracles in this paper. [85, Thereom 41] gives the implementation of the pseudoinverse using QSVT. This can be used to solve the QLSP by applying this pseudoinverse to the quantum state representing the right-hand side.

The recently developed randomization method (RM) [154] is the first algorithm that yields near-optimal scaling with respect to κ, without using techniques such as VTAA. RM was inspired by adiabatic quantum computation (AQC) [79, 9, 97], but relies on the quantum Zeno effect. Both RM and AQC reformulate QLSP into an eigenvalue problem. The runtime complexity of RM is $\mathcal{O}(\kappa\log(\kappa)/\epsilon)$. The recently developed time-optimal AQC(p) and AQC(exp) approaches [15] reduces the runtime complexity to $\mathcal{O}(\kappa/\epsilon)$ and $\mathcal{O}(\kappa\operatorname{polylog}(\kappa/\epsilon))$, respectively. In particular, AQC(exp) achieves the near-optimal complexity with respect to both κ and ϵ, without relying on any amplification procedure. We also remark that numerical observation indicate that the time complexity of the quantum approximate optimization algorithm (QAOA) [78] can be as low as $\mathcal{O}(\kappa\operatorname{polylog}(1/\epsilon))$ [15]. The direct analysis of the complexity of QAOA without relying on the complexity of adiabatic computing (such as AQC(exp)) remains an open question. We demonstrate that quantum eigenstate filtering provides a more versatile approach to obtain the near optimal complexity for solving QLSP. In particular, it can be used to reduce the complexity with respect to ϵ for both adiabatic computing and quantum Zeno effect based methods. In Table 4.1 we compare these aforementioned algorithms in terms of the number of queries to the block-encoding of A. We note that these algorithms rely on different input models but they can all be slightly modified to use the block-encoding assumed in this work.

Recently quantum-inspired classical algorithms based on ℓ^2-norm sampling assumptions [160, 159] have been developed that are only up to polynomially slower than the corresponding quantum algorithms. Similar techniques have been applied to solve low-rank linear systems [53, 84], which achieve exponential speedup in the dependence on the problem size compared to the traditional classical algorithms for the same problem. However, it is unclear whether the classical ℓ^2-norm sampling can be done 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 [22]. Most importantly, the assumption of low-rankness is crucial in these algorithms. Our work is based on the block-encoding model, which could be used to efficiently represent low-rank as well as full-rank matrices on a quantum computer.

Notations: In this paper we use the following asymptotic notations besides the usual $\mathcal{O}$ notation: we write $f=\Omega(g)$ if $g=\mathcal{O}(f)$; $f=\Theta(g)$ if $f=\mathcal{O}(g)$ and $g=\mathcal{O}(f)$; $f=\widetilde{\mathcal{O}}(g)$ if $f=\mathcal{O}(g\operatorname{polylog}(g))$.

We use $\|\cdot\|$ to denote vector or matrix 2-norm: when v is a vector we denote by $\|v\|$ its 2-norm, and when A is matrix we denote by $\|A\|$ its operator norm. For two quantum states $|x\rangle$ and $|y\rangle$, we sometimes write $|x,y\rangle$ to denote $|x\rangle|y\rangle$. We use fidelity to measure how close to each other two quantum states are. Note there are two common definitions for the fidelity between two pure states $|\phi\rangle$ and $|\varphi\rangle$: it is either $|\langle\phi|\varphi\rangle|$ or $|\langle\phi|\varphi\rangle|^2$. Throughout the paper we use the former definition.

Organization: The rest of the paper is organized as follows. In Section 4.2 we briefly review block-encoding and QSP, as well as using QSP to directly solve QLSP with a non-optimal complexity. In Section 4.3 we introduce the minimax polynomial we are using and our eigenstate filtering method based on it. In Section 4.4 we combine eigenstate filtering with AQC to solve the QLSP. In Section 4.5 we present another method to solve the QLSP using QZE and eigenstate filtering. In Section 4.6 we discuss some practical aspects of our algorithms and future work.

4.2 Block-encoding and quantum signal processing

For simplicity we assume $N = 2^n$. An $(m+n)$-qubit unitary operator U is called an (α, m, ϵ)-block-encoding of an n-qubit operator A, if

$$\|A - \alpha(\langle 0^m| \otimes I)U(|0^m\rangle \otimes I)\| \leq \epsilon. \tag{4.1}$$

Another way to express eq. (4.1) is

$$U = \begin{pmatrix} \widetilde{A}/\alpha & * \\ * & * \end{pmatrix},$$

where $*$ can be any block matrices of the correct size and $\|\widetilde{A} - A\| \leq \epsilon$. For instance, when $m = 1$, $\widetilde{A}/\alpha$ is an n-qubit matrix at the upper-left diagonal block of the $(n+1)$-qubit unitary matrix U. Note that the fact $\widetilde{A}/\alpha$ is the upper-left block of a unitary matrix implies $\|\widetilde{A}/\alpha\| \leq \|U\| = 1$. Therefore $\|\widetilde{A}\| \leq \alpha$. Many matrices used in practice can be efficiently block-encoded. For instance, if all entries of A satisfies $|A_{ij}| \leq 1$, and A is Hermitian and d-sparse (i.e. each row / column of A has no more than d nonzero entries), then A has a $(d, n+2, 0)$-encoding U. See [59, Section 4.1] and [34, Lemma 10] for details, as well as [85, Lemma 48] for a more general treatment of sparse matrices.

With a block-encoding available, QSP allows us to construct a block-encoding for an arbitrary polynomial eigenvalue transformation of A.

Theorem 19. ***(Polynomial eigenvalue transformation via quantum signal processing[1] [86, Theorem 31]):*** *Let U be an (α, m, ϵ)-block-encoding of a Her-*

[1] Throughout the paper we use the term QSP to refer to this type of polynomial eigenvalue transformation as well.

mitian matrix A. Let $P \in \mathbb{R}[x]$ be a degree-ℓ real polynomial and $|P(x)| \leq 1/2$ for any $x \in [-1,1]$. Then there exists a $(1, m+2, 4\ell\sqrt{\epsilon}/\alpha)$-block-encoding $\widetilde{U}$ of $P(A/\alpha)$ using ℓ queries of U, $U^\dagger$, and $\mathcal{O}((m+1)\ell)$ other primitive quantum gates.

We remark that Theorem 19 does not meet all our needs because of the constraint $|P(x)| \leq 1/2$. This requirement comes from decomposing the polynomial into the sum of an even and an odd polynomial and then summing them up. When $P(x)$ naturally has a parity this requirement becomes redundant. This enables us to get rid of 1 ancilla qubit. Also for simplicity we assume the block-encoding of A is exact. Therefore we have the following theorem, which can be proved directly from [86, Theorem 2 and Corollary 11].

Theorem 1'. ***(Polynomial eigenvalue transformation with definite parity via quantum signal processing)*** *Let U be an $(\alpha, m, 0)$-block-encoding of a Hermitian matrix A. Let $P \in \mathbb{R}[x]$ be a degree-ℓ even or odd real polynomial and $|P(x)| \leq 1$ for any $x \in [-1,1]$. Then there exists a $(1, m+1, 0)$-block-encoding $\widetilde{U}$ of $P(A/\alpha)$ using ℓ queries of U, $U^\dagger$, and $\mathcal{O}((m+1)\ell)$ other primitive quantum gates.*

Compared to methods such as LCU, one distinct advantage of QSP is that the number of extra ancilla qubits needed is only 1 as shown in Theorem 1'. Hence QSP may be possibly carried out efficiently on intermediate-term devices. Furthermore, a polynomial can be expanded into different basis functions as $P(x) = \sum_{k=0}^{\ell} c_k f_k(x)$, where f_k can be the monomial x^k, the Chebyshev polynomial $T_k(x)$, or any other polynomial. The performance of LCU crucially depends on the 1-norm $\|c\|_1 = \sum_{k=0}^{\ell} |c_k|$, which can be very different depending on the expansion [59]. The block encoding $\widetilde{U}$ in QSP is independent of such a choice, and therefore provides a more *intrinsic* representation of matrix function. We also remark that in QSP, the construction of the block-encoding $\widetilde{U}$ involves a sequence of parameters called phase factors. For a given polynomial $P(x)$, the computation of the phase factors can be efficiently performed on classical computers [92, 85]. There are however difficulties in computing such phase factors, which will be discussed in Section 4.6. For simplicity we assume that the phase factors are given and computed without error.

As an example, we demonstrate how to use QSP to solve QLSP with a Hermitian coefficient matrix A, given by its $(\alpha, m, 0)$-block-encoding U_A. We assume that A, b are normalized as

$$\|A\| = 1, \quad \langle b|b\rangle = 1.$$

We also assume A is Hermitian, and therefore all the eigenvalues of A are real. General matrices can be treated using the standard matrix dilation method (see

Appendix 4.10). Due to the normalization condition, the block-encoding factor satisfies $\alpha \geq \|A\| = 1$. Furthermore, since $\kappa = \|A\|\|A^{-1}\| = \|A^{-1}\|$, the smallest singular value of A is $1/\kappa$. Hence the eigenvalues of A/α are contained in the set $[-1/\alpha, -1/(\alpha\kappa)] \cup [1/(\alpha\kappa), 1/\alpha] \subseteq \mathcal{D}_{1/(\alpha\kappa)}$, where

$$\mathcal{D}_\delta := [-1, -\delta] \cup [\delta, 1].$$

Later we will keep using this notation $\mathcal{D}_\delta$ to denote sets of this type. We first find a polynomial $P(x)$ satisfying $|P(x)| \leq 1$ for any $x \in [-1, 1]$, and $|P(x) - 1/(cx)| \leq \epsilon'$ on $\mathcal{D}_{1/(\alpha\kappa)}$ for $c = 4\alpha\kappa/3$. Note that ϵ' is the accuracy of the polynomial approximation, so that the unnormalized state $P(A/\alpha)\left|b\right\rangle$ would differ from the desired $(\alpha/c)A^{-1}\left|b\right\rangle$ by ϵ'. In order to obtain a normalized solution $P(A/\alpha)\left|b\right\rangle / \|P(A/\alpha)\left|b\right\rangle\|$ that is ϵ-close to the normalized solution $\left|x\right\rangle = A^{-1}\left|b\right\rangle / \|A^{-1}\left|b\right\rangle\|$, we first note that $\|A^{-1}\left|b\right\rangle\| \geq 1$. So the normalization would amplify the error by a factor of approximately $c/(\alpha\|A^{-1}\left|b\right\rangle\|) \leq 4\kappa/3$. Therefore we may choose $\epsilon' = 3\epsilon/4\kappa$. Then we can find an odd polynomial of degree $\mathcal{O}(\alpha\kappa\log(\kappa/\epsilon))$, where ϵ is the desired precision, satisfying this by [85, Corollary 69]. Then by Theorem 1' we have a circuit $\widetilde{U}$ satisfying

$$\begin{aligned}\widetilde{U}\left|0^{m+1}\right\rangle\left|b\right\rangle &= \left|0^{m+1}\right\rangle\left(P(A/\alpha)\left|b\right\rangle\right) + \left|\phi\right\rangle \\ &\approx \left|0^{m+1}\right\rangle\left(\frac{\alpha}{c}A^{-1}\left|b\right\rangle\right) + \left|\phi\right\rangle,\end{aligned}$$

where $\left|\phi\right\rangle$ is orthogonal to all states of the form $\left|0^{m+1}\right\rangle\left|\psi\right\rangle$. Measuring the ancilla qubits, we obtain the a normalized quantum state $P(A/\alpha)\left|b\right\rangle / \|P(A/\alpha)\left|b\right\rangle\|$ that is ϵ-close to the normalized solution $\left|x\right\rangle$ with probability $\Theta\left(\left(\frac{\alpha}{c}\|A^{-1}\left|b\right\rangle\|\right)^2\right)$.

As $\|A^{-1}\left|b\right\rangle\| \geq 1$, the probability of success is $\Omega(1/\kappa^2)$. Using amplitude amplification [42], the number of repetitions needed for success can be improved to $\mathcal{O}(\kappa)$. Furthermore, the query complexity of application of $\widetilde{U}$ is $\mathcal{O}(\alpha\kappa\log(\kappa/\epsilon))$. Therefore the overall query complexity is $\mathcal{O}(\alpha\kappa^2\log(\kappa/\epsilon))$.

We observe that the quadratic scaling with respect to κ is very much attached to the procedure above: each application of QSP costs $\mathcal{O}(\kappa)$ queries of $U, U^\dagger$, and the other from that QSP needs to be performed for $\mathcal{O}(\kappa)$ times. The same argument applies to other techniques such as LCU. To reduce the κ complexity along this line, one must modify the procedure substantially to avoid the multiplication of the two κ factors, such as using the modified LCU based on VTAA [59].

4.3 Eigenstate filtering using a minimax polynomial

Now consider a Hermitian matrix H, with a known eigenvalue λ that is separated from other eigenvalues by a gap Δ. H is assumed to have an $(\alpha, m, 0)$-block-encoding denoted by U_H. We want to preserve the λ-eigenstate while filtering out all other eigenstates. Let P_λ denote the projection operator into the λ-eigenspace of H. The basic idea is, suppose we have a polynomial P such that $P(0) = 1$ and $|P(x)|$ is small for $x \in \mathcal{D}_{\Delta/(2\alpha)}$, where we use the notation $\mathcal{D}_\delta = [-1, -\delta] \cup [\delta, 1]$ that has been introduced earlier, then $P((H - \lambda I)/(\alpha + |\lambda|)) \approx P_\lambda$. This is the essence of the algorithm we are going to introduce below. The reason we need to introduce the factors 2α and $\alpha + |\lambda|$ is that the block-encoding of $H - \lambda I$ will involve a factor $\alpha + |\lambda|$, and this is explained in detail in Appendix 4.7. Since $|\lambda| \leq \alpha$ by definition of the operator norm, we have $\alpha + |\lambda| \leq 2\alpha$. Therefore when λ is separated from the rest of the spectrum of H by a gap Δ, 0 is separated from the rest of the spectrum of $(H - \lambda I)/(\alpha + |\lambda|)$ by a gap $\Delta/(\alpha + |\lambda|) \geq \Delta/(2\alpha) = \widetilde{\Delta}$.

We use the following 2ℓ-degree polynomial

$$R_\ell(x; \Delta) = \frac{T_\ell\left(-1 + 2\frac{x^2 - \Delta^2}{1 - \Delta^2}\right)}{T_\ell\left(-1 + 2\frac{-\Delta^2}{1 - \Delta^2}\right)},$$

where $T_\ell(x)$ is the ℓ-th Chebysehv polynomial of the first kind. This polynomial is inspired by the shifted and rescaled Chebyshev polynomial discussed in [143, Theorem 6.25]. A plot of the polynomial is given in Fig. 4.1. $R_\ell(x; \Delta)$ has several nice properties:

Lemma 20. *(i) $R_\ell(x; \Delta)$ solves the minimax problem*

$$\underset{p(x) \in \mathbb{P}_{2\ell}[x], p(0)=1}{\text{minimize}} \ \max_{x \in \mathcal{D}_\Delta} |p(x)|.$$

(ii) $|R_\ell(x; \Delta)| \leq 2e^{-\sqrt{2}\ell\Delta}$ for all $x \in \mathcal{D}_\Delta$ and $0 < \Delta \leq 1/\sqrt{12}$. Also $R_\ell(0; \Delta) = 1$.
(iii) $|R_\ell(x; \Delta)| \leq 1$ for all $|x| \leq 1$.

A proof of the above lemma is provided in Appendix 4.11. If we apply this polynomial to $H - \lambda I$, Lemma 20 (i) states that R_ℓ achieves the best compression ratio of the unwanted components, among all polynomials of degrees up to 2ℓ. To prepare a quantum circuit, we define $\widetilde{H} = (H - \lambda I)/(\alpha + |\lambda|)$. Then we can also construct a $(1, m+1, 0)$-block-encoding for $\widetilde{H}$ (see Appendix 4.7). The gap separating 0 from other eigenvalues of $\widetilde{H}$ is lower bounded by $\widetilde{\Delta} = \Delta/2\alpha$, as explained at the

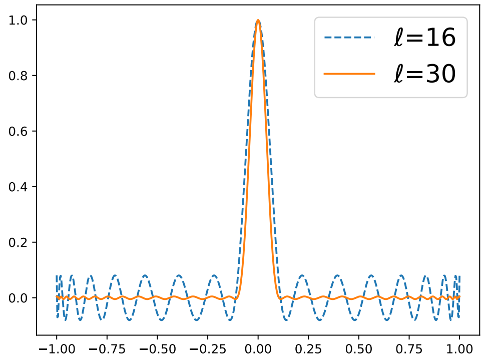


Figure 4.1: The polynomial $R_\ell(x,\Delta)$ for $\ell=16$ and 30, $\Delta=0.1$.

beginning of this section. Together with the fact that $\|\widetilde{H}\|\leq 1$, we find that the spectrum of $\widetilde{H}$ is contained in $\mathcal{D}_{\widetilde{\Delta}}\cup\{0\}$.

We then apply Lemma 20. Note that the requirement when $\widetilde{\Delta}>1/\sqrt{12}$ might not be satisfied, we can always set $\widetilde{\Delta}=1/\sqrt{12}$ and this does not affect the asymptotic complexity as $\widetilde{\Delta}\to 0$. Because of (ii) of Lemma 20, we have

$$\|R_\ell(\widetilde{H},\widetilde{\Delta})-P_\lambda\|\leq 2e^{-\sqrt{2}\ell\widetilde{\Delta}}.$$

Also because of (iii), and the fact that $R_\ell(x;\widetilde{\Delta})$ is even, we may apply Theorem 1' to implement $R_\ell(\widetilde{H};\widetilde{\Delta})$ using QSP. This gives the following theorem:

Theorem 21. ***(Eigenstate filtering):*** *Let H be a Hermitian matrix and U_H is an $(\alpha,m,0)$-block-encoding of H. If λ is an eigenvalue of H that is separated from the rest of the spectrum by a gap Δ, then we can construct a $(1,m+2,\epsilon)$-block-encoding of P_λ, by $\mathcal{O}((\alpha/\Delta)\log(1/\epsilon))$ applications of (controlled-) U_H and $U_H^\dagger$, and $\mathcal{O}((m\alpha/\Delta)\log(1/\epsilon))$ other primitive quantum gates.*

Suppose we can prepare a state $|\psi\rangle=\gamma|\psi_\lambda\rangle+|\perp\rangle$ using an oracle O_ψ, where $|\psi_\lambda\rangle$ is a λ-eigenvector and $\langle\psi_\lambda|\perp\rangle=0$, for some $0<\gamma\leq 1$. Theorem 21 states that we can get an ϵ-approximation to $|\psi_\lambda\rangle$ with $\mathcal{O}((\alpha/\Delta)\log(1/(\gamma\epsilon)))$ queries to U_H, with a successful application of the block-encoding of P_λ, denoted by U_{P_λ}. The fact we have $1/(\gamma\epsilon)$ instead of $1/\epsilon$ in the logarithm is due to the error amplification going from an unnormalized state to a normalized state, similar to that discussed in the application of QSP to QLSP in Section 4.2. The probability of applying this block-encoding successfully, i.e. getting all 0's when measuring ancilla qubits, is at least γ^2. Therefore when $|\psi\rangle$ can be repeatedly prepared by an oracle, we only need to run U_{P_λ} and the oracle on average $\mathcal{O}(1/\gamma^2)$ times to obtain $|\psi_\lambda\rangle$ successfully. With amplitude amplification we can reduce this number to $\mathcal{O}(1/\gamma)$. However this is not necessary when $\gamma = \Omega(1)$, when without using amplitude amplification we can already obtain $|\psi_\lambda\rangle$ by using the oracle for initial state and U_{P_λ} $\mathcal{O}(1)$ times.

We remark that the eigenstate filtering procedure can also be implemented by alternative methods such as LCU. The polynomial $R_\ell(\cdot, \widetilde{\Delta})$ can be expanded exactly into a linear combination of the first $2\ell + 1$ Chebyshev polynomials. The 1-norm of the expansion coefficients is upper bounded by $2\ell + 2$ because $|R_\ell(x, \widetilde{\Delta})| \leq 1$. However, this comes at the expense of additional $\mathcal{O}(\log \ell)$ qubits needed for the LCU expansion [59].

Besides the projection operator, we can use this filtering procedure to implement many other related operators. First we consider implementing the reflection operator about the target λ-eigenstate (or λ-eigenspace if there is degeneracy), $2P_\lambda - I$, which is useful in the amplitude amplification procedure [90, 42]. This problem has been considered in Ref. [61].

For a given Hamiltonian H, with the same assumptions as in Theorem 21, and $\widetilde{H} = (H - \lambda I)/(\alpha + |\lambda|)$ as constructed above, we define

$$R_\lambda = 2P_\lambda - I,$$

where P_λ is the projection operator into the λ-eigenspace of H. Using a polynomial $S_\ell(x; \Delta)$ constructed from $R_\ell(x; \Delta)$ as introduced in Appendix 4.8, we can implement the reflection operator R_λ through QSP. The cost is summarized as follows:

Theorem 22. *Under the same assumption as Theorem 21, a $(1, m+2, \epsilon)$-block-encoding of R_λ, the reflection operator about the λ-eigenspace of H, can be constructed using $\mathcal{O}((\alpha/\Delta)\log(1/\epsilon))$ applications of (controlled-) U_H and $U_H^\dagger$, and $\mathcal{O}((m\alpha/\Delta)\log(1/\epsilon))$ other primitive quantum gates.*

For the proof see Appendix 4.8. This reflection operator further enables us to construct a block-encoding of the θ-reflection operator.

$$P_\lambda + e^{\mathrm{i}\theta}(I - P_\lambda).$$

This operator is useful in fixed-point amplitude amplification [179, 91]. The cost is summarized as follows:

Corollary 23. *Under the same assumption as Theorem 21, a $(1, m+3, \epsilon)$-block-encoding of $P_\lambda + e^{\mathrm{i}\theta}(I - P_\lambda)$, where P_λ is the projection operator into the λ-eigenspace* *of H, can be constructed using $\mathcal{O}((\alpha/\Delta)\log(1/\epsilon))$ applications of (controlled-) U_H and $U_H^\dagger$, and $\mathcal{O}((m\alpha/\Delta)\log(1/\epsilon))$ other primitive quantum gates.*

The proof can be found in Appendix 4.8.

In this paper we focus on obtaining the eigenstate corresponding to an eigenvalue that is known exactly. If instead of a single known eigenvalue, we want keep all eigenvalues in a certain interval, and filter out the rest, we can use a linear combination of polynomials used to approximate the sign function [86, Lemma 14], together with constant shift. The filtering polynomial for this kind of task can also be obtained numerically through Remez algorithm [139], followed by a optimization based procedure to efficiently identify the phase factors. For more details we refer readers to Ref. [70].

Remark 24. *In the special case where $\|H\| = 1$, the target eigenvalue is 1, and we have access to a $(1, m, 0)$-block-encoding of H, then a quadratically improved dependence on the gap can be achieved using polynomials such as [143, Eq. (6.113)]. This is useful for obtaining the stationary distribution of an ergodic and reversible Markov chain because the discriminant matrix [158, 19, 18] can be block-encoded efficiently in a reflection operator, and its 1-eigenstate is $\sum_j \sqrt{\pi_j}\,|j\rangle$ where $\pi = (\pi_j)$ is the stationary distribution.*

4.4 Solving QLSP: eigenstate filtering with adiabatic quantum computing

To define QLSP, we assume that a d-sparse matrix A can be accessed by oracles $O_{A,1}$, $O_{A,2}$ as

$$O_{A,1}|j,l\rangle = |j,\nu(j,l)\rangle, \quad O_{A,2}|j,k,z\rangle = |j,k,A_{jk}\oplus z\rangle, \tag{4.2}$$

where $j,k,l,z \in [N]$, and $\nu(j,l)$ is the row index of the l-th nonzero element in the j-th column. The right hand side vector $|b\rangle$ can be prepared with an oracle O_B as

$$O_B\,|0\rangle = |b\rangle\,. \tag{4.3}$$

This is the same as the assumption used in [59, 154]. The oracles can be used to construct a $(d, n+2, 0)$-block-encoding of A [34, 59].

We assume the singular values of A are contained in $[1/\kappa, 1]$ for some $\kappa > 1$. Therefore κ here is an upper bound for the condition number, which is defined as the ratio between the largest and the smallest singular values. It is thus guaranteed that when A is Hermitian its eigenvalues are contained in $\mathcal{D}_{1/\kappa} = [-1, -1/\kappa] \cup [1/\kappa, 1]$. In Ref. [59] it is assumed that that $\|A\| = 1$ and the condition number is exactly κ [59, Problem 1], which is slightly stronger than the assumption we are currently using.

Remark 25. *We can always assume without loss of generality that A is Hermitian. Because when A is not Hermitian we can solve an extended linear system as described in Appendix 4.10, where the coefficient matrix is*

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$$

This is a Hermitian matrix, and when A is d-sparse, this matrix is d-sparse as well. If A has singular values $\{\sigma_k\}$, then the dilated Hermitian matrix has real eigenvalues $\{\pm\sigma_k\}$. Therefore the two matrices have the same condition number, and when the singular values of A are contained in $[1/\kappa, 1]$ the spectrum of the above dilated matrix is contained in $[-1, -1/\kappa] \cup [1/\kappa, 1]$.

We will then apply the method we developed in the last section to QLSP. To do this we need to convert QLSP into an eigenvalue problem. For simplicity we assume A is Hermitian positive-definite. The indefinite case is addressed in Appendix 4.10, which uses different Hamiltonians but only requires minor modifications. We define

$$H_1 = \begin{pmatrix} 0 & AQ_b \\ Q_b A & 0 \end{pmatrix} = |0\rangle\langle 1| \otimes AQ_b + |1\rangle\langle 0| \otimes Q_b A, \tag{4.4}$$

where $Q_b = I - |b\rangle\langle b|$. This Hamiltonian has been used in Refs. [154, 15]. As discussed in Appendix 4.7, we can construct a $(d, n+4, 0)$-block-encoding of H_1, denoted by U_{H_1} by applying $O_B, O_{A,1}, O_{A,2}$ twice.

We may readily verify that the 0-eigenspace, i.e. the null space, of H_1 is spanned by $|0\rangle|x\rangle = (x, 0)^\top$, where $|x\rangle$ is the solution, i.e. $A|x\rangle \propto |b\rangle$, and $|1\rangle|b\rangle = (0, b)^\top$, by considering the null space of H_1^2. The rest of the spectrum is separated from 0 by a gap of $1/\kappa$ [15, 154]. Therefore to apply the eigenstate filtering method, we only need an initial state[2] with non-vanishing overlap with the target eigenstate $|0\rangle|x\rangle$ that can be efficiently prepared. We will prepare this initial state using the time-optimal adiabatic quantum computing.

[2] We will later discuss how to use AQC and QZE to prepare this state. Therefore it is worth pointing out that by initial state here we mean the state on which we apply the eigenstate filtering, rather than the initial state of AQC or QZE.

Choosing the eigenpath

To use adiabatic quantum computing we need to first specify the eigenpath we are going to follow. We define

$$H_0 = \begin{pmatrix} 0 & Q_b \\ Q_b & 0 \end{pmatrix} = \sigma_x \otimes Q_b. \tag{4.5}$$

and

$$H(f) = (1-f)H_0 + fH_1,$$

where H_1 is defined in Eq. (4.4).

We will then evolve the system following the 0-eigenstates of each $H(f)$. These eigenstates form an eigenpath linking the initial state to the solution to the linear system. There are several important properties of the Hamiltonians $H(f)$ and of the eigenpath which we discuss below, though some of them we will only use in the algorithm based on the quantum Zeno effect.

The null space of $H(f)$ is two-dimensional, and we will pay special attention to this fact in our analysis. The non-zero eigenvalues of $H(f)$ appear in pairs. Let $\lambda_j(f)$, $j = 1, 2, \ldots, N-1$ be all the positive eigenvalues of $H(f)$, and $|z_j(f)\rangle$ be the corresponding eigenvectors, then we may readily check

$$H(f)(\sigma_z \otimes I)\,|z_j(f)\rangle = -\lambda_j(f)(\sigma_z \otimes I)\,|z_j(f)\rangle\,.$$

Therefore $-\lambda_j(f)$ is also an eigenvalue of $H(f)$ with corresponding eigenvector $(\sigma_z \otimes I)\,|z_j(f)\rangle$, for $j = 1, 2, \ldots, N-1$. Thus we have obtained all the non-zero eigenvalues and corresponding eigenvectors.

The form of the matrices in Eqs. (4.4) and (4.5) is important for achieving $\mathcal{O}(\kappa)$ complexity in our algorithms because they ensure the gap between 0 and other eigenvalues for all f is lower bounded by

$$\Delta_*(f) = 1 - f + \frac{f}{\kappa}. \tag{4.6}$$

A proof can be found in [15].

Now we are ready to specify the eigenpath. For any f, we let $|x(f)\rangle$ be some vector such that

$$((1-f)I + fA)\,|x(f)\rangle \propto |b\rangle\,. \tag{4.7}$$

We can then see that the null space of $H(f)$ is spanned by $|\bar{x}(f)\rangle = |0\rangle\,|x(f)\rangle$ and $|1\rangle\,|b\rangle$. This requirement pins down the choice for $|x(f)\rangle$ up to a time-dependent global phase. By requiring the phase to be geometric, i.e.

$$\langle x(f)|\partial_f|x(f)\rangle = 0, \tag{4.8}$$

the eigenpath $\{|x(f)\rangle\}$ becomes uniquely defined when we require $|x(0)\rangle = |b\rangle$. Note the above equation is slightly problematic in that we do not know beforehand that $|x(f)\rangle$ is differentiable. However this turns out not to be a problem because we can establish the differentiability in Appendix 4.12. Furthermore, we have the estimate

$$\|\partial_f |x(f)\rangle\| \leq \frac{2}{\Delta_*(f)}. \tag{4.9}$$

The derivation of the existence and uniqueness of the differentiable eigenpath, together with the estimate (4.9) are given in Appendix 4.12.

An important quantity we need to use in our analysis is the eigenpath length

$$L = \int_0^1 \|\partial_f |x(f)\rangle\| \mathrm{d}f,$$

and by (4.9) we have

$$L \leq \int_0^1 \frac{2}{\Delta_*(f)} \mathrm{d}f = \frac{2\log(\kappa)}{1 - 1/\kappa}. \tag{4.10}$$

We also define the eigenpath length $L(a, b)$ between $0 < a < b < 1$ and it is bounded by

$$L(a,b) = \int_a^b \|\partial_f |x(f)\rangle\| \mathrm{d}f \leq \frac{2}{1-1/\kappa} \log\left(\frac{1-(1-1/\kappa)a}{1-(1-1/\kappa)b}\right) =: L_*(a,b). \tag{4.11}$$

Time-optimal adiabatic quantum computing

Here we briefly review the procedure of solving QLSP using the recently developed time-optimal AQC [15] and the eigenpath described in the previous section that has been used in [15, 154].

As noted before, the null space of $H(f)$ is two-dimensional, which contains an unwanted 0-eigenvector $|1\rangle |b\rangle = (0, b)^\top$. However this 0-eigenvector is not accessible in the AQC time-evolution

$$\frac{1}{T} \mathrm{i}\partial_s |\psi_T(s)\rangle = H(f(s)) |\psi_T(s)\rangle, \quad |\psi_T(0)\rangle = |0\rangle |b\rangle,$$

for scheduling function $f : [0, 1] \to [0, 1]$, which is a strictly increasing mapping with $f(0) = 0, f(1) = 1$. We find that

$$(\langle 1| \langle b|) |\psi_T(s)\rangle = 0,$$

for all $s \in [0,1]$. This is due to

$$\frac{1}{T}\mathrm{i}\partial_s(\langle 1| \langle b|) |\psi_T(s)\rangle = (\langle 1| \langle b|)H(f(s)) |\psi_T(s)\rangle = 0,$$

and $(\langle 1| \langle b|) |\psi_T(0)\rangle = 0$. This fact gets rid of the problem.

The parameter T needed to reach a certain target accuracy ϵ is called the runtime complexity (or simply the time complexity). The simplest choice for the scheduling function is $f(s) = s$, which gives the "vanilla AQC". Besides $|0\rangle |x\rangle$, all other eigenstates of H_1 that can be connected to $|0\rangle |b\rangle$ through an adiabatic evolution are separated from $|0\rangle |x\rangle$ by an energy gap of at least $1/\kappa$ [15, 154]. The time complexity of vanilla AQC is at least $T \sim \mathcal{O}(\kappa^2/\epsilon)$ [97, 15, 9, 72].

By properly choosing a scheduling function $f(s)$, the time complexity of AQC can be significantly improved. There are two time-optimal scheduling functions proposed in [15]. The first method is called AQC(p). For $1 < p < 2$, AQC(p) adopts the schedule

$$f(s) = \frac{\kappa}{\kappa - 1}\left[1 - \left(1 + s(\kappa^{p-1} - 1)\right)^{\frac{1}{1-p}}\right]. \tag{4.12}$$

This reduces the time complexity to $\mathcal{O}(\kappa/\epsilon)$, which is optimal for κ, but the scaling with respect to ϵ is the same. The second method is called AQC(exp), which uses a different scheduling function to achieve time complexity $\mathcal{O}\left(\kappa \log^2(\kappa) \log^4\left(\frac{\log \kappa}{\epsilon}\right)\right)$.

All AQC methods are time-dependent Hamiltonian simulation problem, which can be implemented using e.g. truncated Dyson series for simulating the time-dependent Hamiltonian [116]. Although AQC(exp) scales near-optimally with respect to κ and ϵ, numerical evidence indicates that the preconstant of AQC(exp) can be higher than AQC(p). Hence when a low accuracy $\epsilon \sim \mathcal{O}(1)$ is needed, AQC(p) can require a smaller runtime in practice. In the discussion below, we will consider AQC(p).

The details of the time-dependent Hamiltonian simulation for AQC are discussed in Appendix 4.9, and the query complexity for implementing AQC(p) on a gate-based quantum computer is $\widetilde{\mathcal{O}}(\kappa/\epsilon)$.

Improved dependence on ϵ

We now use eigenstate filtering to accelerate AQC(p) and reduce the query complexity to $\log(1/\epsilon)$. As mentioned before, once we have access to H_1 defined in (4.4), through the block-encoding U_{H_1} constructed in Appendix 4.7 we only need an initial state for eigenstate filtering (note that this is not the initial state of the AQC time-evolution):

$$|\widetilde{x}_0\rangle = \gamma_0 |0\rangle |x\rangle + \gamma_1 |1\rangle |b\rangle + |\perp\rangle \tag{4.13}$$

with $|\gamma_0| = \Omega(1)$ and $|\perp\rangle$ orthogonal to the null space. The initial state $|\widetilde{x}_0\rangle$ can be prepared using the time-optimal AQC procedure. Again we first assume A is Hermitian positive definite. To make $|\gamma_0| = \Omega(1)$ we only need to run AQC(p) to constant precision, and thus the linear dependence on precision is no longer a problem. Therefore the time complexity of AQC(p) is $\mathcal{O}(\kappa)$. However we still need to implement AQC(p) on a quantum circuit. To do this we use the time-dependent Hamiltonian simulation introduced in [116], which gives a $\mathcal{O}(d\kappa \log(d\kappa)/\log\log(d\kappa))$ query complexity to achieve $\mathcal{O}(1)$ precision, for a d-sparse matrix A. This procedure also needs to be repeated $\mathcal{O}(1)$ times. It should be noted that γ_1 in Eq. (4.13) comes entirely from the error of the Hamiltonian simulation, since AQC should ensure that the state is orthogonal to $|1\rangle|b\rangle$ for all t. Details on performing this time-dependent Hamiltonian simulation is given in Appendix 4.9.

Then we can run the eigenstate filtering algorithm described in Section 4.3 to precision ϵ to obtain $R_\ell(H_1/d; 1/(d\kappa))|\widetilde{x}_0\rangle$. The $|\perp\rangle$ component will be filtered out, while the $|0\rangle|x\rangle$ and $|1\rangle|b\rangle$ components remain. To further remove the $|1\rangle|b\rangle$ component, we measure the first qubit. Upon getting an outcome 0, the outcome state will just be $|0\rangle|x\rangle + \mathcal{O}(\epsilon)$. The success probability of applying the eigenstate filtering is lower bounded by $|\gamma_0|^2 + |\gamma_1|^2$, and the success probability of obtaining 0 in measurement is $|\gamma_0|^2/(|\gamma_0|^2 + |\gamma_1|^2) + \mathcal{O}(\epsilon)$. Thus the total success probability is $\Omega(1)$. Each single application of eigenstate filtering applies U_{H_1}, and therefore $O_{A,1}$, $O_{A,2}$, and O_B, for $\mathcal{O}(d\kappa \log(1/\epsilon))$ times. It only needs to be repeated $\Omega(1)$ times so the total query complexity of eigenstate filtering is still $\mathcal{O}(d\kappa \log(1/\epsilon))$.

In eigenstate filtering we need $\mathcal{O}(nd\kappa \log(1/\epsilon))$ additional primitive gates as mentioned in Theorem 21. In time-dependent Hamiltonian simulation the addition number of primitive gates needed is $\mathcal{O}(d\kappa(n + \log(d\kappa))\frac{\log(d\kappa)}{\log\log(d\kappa)})$. Both procedures are repeated $\mathcal{O}(1)$ times and therefore in total we need $\mathcal{O}\left(d\kappa\left(n\log(\frac{1}{\epsilon}) + (n + \log(d\kappa))\frac{\log(d\kappa)}{\log\log(d\kappa)}\right)\right)$ additional primitive gates.

The number of qubits needed in the eigenstate filtering procedure using QSP is $\mathcal{O}(n)$ which mainly comes from the original size of the problem and block-encoding. Extra ancilla qubits introduced as a result of eigenstate filtering is only $\mathcal{O}(1)$. In the Hamiltonian simulation $\mathcal{O}(n + \log(d\kappa))$ qubits are needed (see Appendix 4.9). Therefore the total number of qubits needed is $\mathcal{O}(n + \log(d\kappa))$.

The procedure above can be generalized to Hermitian indefinite matrices, and general matrices that are not necessarily Hermitian (see Appendix 4.10). As discussed in Remark 25, for general matrices we should assume the singular values instead of eigenvalues of A are contained in $[1/\kappa, 1]$. Therefore our QLSP solver can be summarized as

Theorem 26. *A is a d-sparse matrix whose singular values are in $[1/\kappa, 1]$ and can be queried through oracles $O_{A,1}$ and $O_{A,2}$ in* (4.2)*, and $|b\rangle$ is given by an oracle O_B in* (4.3)*. Then $|x\rangle \propto A^{-1}|b\rangle$ can be obtained with fidelity $1-\epsilon$, succeeding with probability $\Omega(1)$ with ancilla qubits measurement outcome indicating success, using*

1. $\mathcal{O}\left(d\kappa(\frac{\log(d\kappa)}{\log\log(d\kappa)} + \log(\frac{1}{\epsilon}))\right)$ *queries to $O_{A,1}$, $O_{A,2}$, and O_B,*
2. $\mathcal{O}\left(d\kappa\left(n\log(\frac{1}{\epsilon}) + (n+\log(d\kappa))\frac{\log(d\kappa)}{\log\log(d\kappa)}\right)\right)$ *other primitive gates,*
3. $\mathcal{O}(n+\log(d\kappa))$ *qubits.*

When the gate complexity of $O_{A,1}$, $O_{A,2}$, and O_B are poly(n) the total gate complexity, and therefore runtime, by the above theorem, will be $\widetilde{\mathcal{O}}(\text{poly}(n)d\kappa\log(1/\epsilon))$.

Remark 27. *Although in total we need $\mathcal{O}(n+\log(d\kappa))$ ancilla qubits, only $\mathcal{O}(\log(d\kappa))$ comes sources other than the block-encoding of A. In other words, our method only adds $\mathcal{O}(\log(d\kappa))$ ancilla qubits to those that are unavoidable as long as we use this way of block-encoding of a sparse A. These extra ancilla qubits are mainly a result of using time-dependent Hamiltonian simulation. Also, although in the theorem we assumed A is a sparse matrix, we have only used this fact to build its block-encoding. Given the block-encoding of a matrix A that is not necessarily sparse, the above procedure can still be carried out directly. This is also true for Theorem 29 which we are going to introduce later.*

We present numerical results obtained on a classical computer in Fig. 4.2 to validate the complexity estimate. In the numerical test, we solve the linear system $A|x\rangle \propto |b\rangle$, where A is formed by adding a randomly generated symmetric positive definite tridiagonal matrix B, whose smallest eigenvalue is very close to 0, to a scalar multiple of the identity matrix. After properly rescaling, the eigenvalues of A lie in $[-1, 1]$. This construction enables us to estimate condition number with reasonable accuracy without computing eigenvalues. The off-diagonal elements of B are drawn uniformly from $[-1, 0]$ and the diagonal elements are the negative of sums of two adjacent elements on the same row. The $(0, 0)$ and $(N-1, N-1)$ elements of B are slightly larger so that B is positive definite. $|b\rangle$ is drawn from the uniform distribution on the unit sphere.

With A and $|b\rangle$ chosen, we first run the AQC time evolution for time $\mathcal{O}(\kappa)$ as described at the beginning of this section, and then apply eigenstate filtering using the polynomial $R_\ell(x; 1/d\kappa)$ with degree 2ℓ. Denoting the resulting quantum state by $|\widetilde{x}\rangle$ we then compute the fidelity $\eta = |\langle x|\widetilde{x}\rangle|$. Fig. 4.2 shows the relation between η, κ, and ℓ obtained in the numerical experiment.

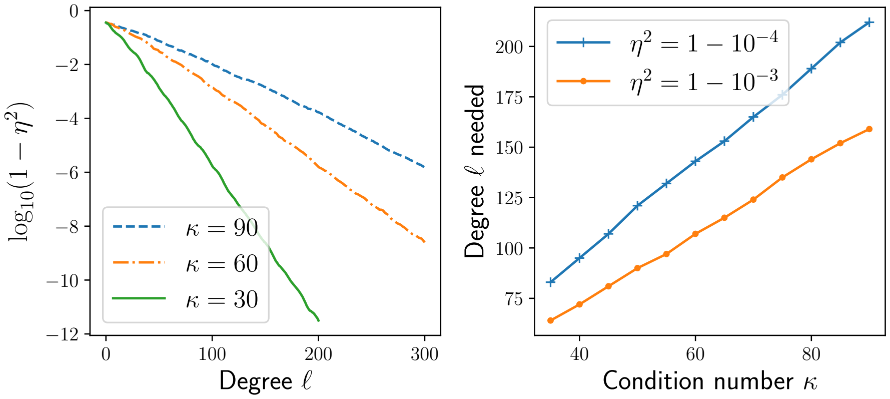


Figure 4.2: Left: fidelity η converges to 1 exponentially as ℓ in the eigenvalues filtering algorithm increases, for different κ. Right: the smallest ℓ needed to achieve fixed fidelity η grows linearly with respect to condition number κ. The initial state in eigenstate filtering is prepared by running AQC(p) for $T = 0.2\kappa$, with $p = 1.5$, which achieves an initial fidelity of about 0.6.

4.5 Solving QLSP: eigenstate filtering with quantum Zeno effect

Quantum Zeno effect (QZE) is the phenomenon that frequent measurements hinders a quantum system's transition from its initial state to other states [121, 77, 76, 29, 45]. A variant of QZE [39, Lemma 1] can be viewed as a particular way for implementing adiabatic quantum computing [136, 150, 109], and this is what we mean by QZE throughout this work unless stated otherwise. The basic idea of this variant of QZE is to follow an adiabatic path through repeated measurement, which acts as projection operators to the instantaneous eigenstate along the adiabatic path. This inspired the randomization method for performing computation based on QZE [39, 154].

In the context of solving QLSP, again for simplicity we first assume A is Hermitian positive definite. Instead of running time-dependent Hamiltonian simulation to evolve from the 0-eigenstate of H_0 to the 0-eigenstate of H_1, we consider applying a series of projections to traverse the eigenpath. Choosing $0 = f_0 < f_1 < \ldots < f_M = 1$, for each $j = 0, 1, \ldots, M-1$, we start from the 0-eigenstate $|0\rangle\,|x(f_j)\rangle$ of $H(f_j)$, where $|x(f)\rangle$ is defined in Eqs. (4.7) and (4.8), and project into the null space of $H(f_{j+1})$.

In the end we obtain the 0-eigenstate of $H(1) = H_1$. This is essentially the same as performing projective measurement for each j [57, 39, 154]. If the projective measurements are done approximately using quantum phase estimation or phase randomization, there will be a linear dependence on $1/\epsilon$ in runtime, ϵ being the desired precision.

In this section we combine eigenstate filtering with Zeno-based computation to reduce the error dependence from $\mathcal{O}(1/\epsilon)$ to $\mathcal{O}(\log(1/\epsilon))$, thanks to the possibility of performing approximate projections with high precision. However, several issues demand our attention in the procedure outlined at the beginning of this section. First, we need to specify the choice of $\{f_j\}$, which plays an important role in the lower bound of M needed to ensure at least constant success probability. Second, the null space of each $H(f_j)$ is 2-dimensional. Therefore the eigenpath is not unique, and we need to specify the eigenpath we are going to traverse, which has been done in Sec. 4.4, and to ensure the undesired part of the null space does not interfere with our computation.

The algorithm

As in Section 4.4, the goal is to produce a state close to the solution state $|x\rangle$ of the QLSP with fidelity at least $1-\epsilon$ for some given $0 < \epsilon < 1$. In this section we describe the procedure of the Quantum Zeno effect state preparation. We need to choose a scheduling function

$$f(s) = \frac{1-\kappa^{-s}}{1-\kappa^{-1}} \tag{4.14}$$

and define $f_j = f(s_j)$ where $s_j = j/M$. Without the scheduling we will end up with an unfavorable square dependence on the minimum spectral gap along the eigenpath [57]. This scheduling is chosen so that

$$L(f_j, f_{j+1}) \leq L_*(f_j, f_{j+1}) = \frac{2\log(\kappa)}{M(1-1/\kappa)}, \tag{4.15}$$

which implies we are dividing the interval $[0, 1]$ of f into M segments of equal L_*-length.

Before we describe the algorithm we need to first introduce some notations and block-encodings we need to use. From the block-encoding of H_0 and H_1 described in Appendix 4.7, we can construct $(1-f+fd, n+6, 0)$-block-encoding for each $H(f)$, denoted by $U_H(f)$. This construction uses [86, Lemma 29], through

$$H(f) = (1-f+fd)\left(\langle c| \otimes I\right)\left[|0\rangle\langle 0| \otimes H_0 + |1\rangle\langle 1| \otimes (H_1/d)\right]\left(|c\rangle \otimes I\right),$$

where

$$|c\rangle = \frac{1}{\sqrt{1-f+fd}}(\sqrt{1-f}\,|0\rangle + \sqrt{fd}\,|1\rangle).$$

We need to use H_1/d instead of H_1 because there is a d factor involved in the block-encoding of H_1 (see Appendix 4.7) , and the above equation shows we get a $1-f+fd$ factor in the block-encoding of $H(f)$ because we need to normalize the coefficient vector $|c\rangle$. For a more detailed discussion see Appendix 4.7. Applying the eigenstate filtering procedure in Section 4.3 to precision ϵ_P gives us an $(1, n+7, \epsilon_P)$-block-encoding of

$$\bar{P}_0(f) = |0\rangle\,|x(f)\rangle\,\langle 0|\,\langle x(f)| + |1\rangle\,|b\rangle\,\langle 1|\,\langle b|\,, \tag{4.16}$$

which we denote by $U_{P_0}(f)$. By Theorem 21 this uses $U_H(f)$ and its inverse $\mathcal{O}(\frac{d}{\Delta_*(f)}\log(\frac{1}{\epsilon_P}))$ times. Note that one ancilla qubit introduced in Theorem 21 is redundant because we do not need to shift by a multiple of the identity matrix. By definition of block-encoding we have

$$\left\|\bar{P}_0(f) - (\langle 0^{n+7}| \otimes I_{n+1})U_{P_0}(f)(|0^{n+7}\rangle \otimes I_{n+1})\right\| \le \epsilon_P.$$

Here for clarity we use I_r to denote the identity operator acting on r qubits. Note that we need access to

$$P_0(f) = |x(f)\rangle\,\langle x(f)|\,, \tag{4.17}$$

which is the projection operator onto $|x(f)\rangle$, instead of $\bar{P}_0(f)$, which is the projection operator onto $|0\rangle\,|x(f)\rangle$. We now consider how to approximate $P_0(f)$. Because of the fact

$$P_0(f) = (\langle 0| \otimes I_n)\bar{P}_0(f)(|0\rangle \otimes I_n),$$

we denote

$$\widetilde{P}_0(f) = (\langle 0^{n+7}|\,\langle 0| \otimes I_n)U_{P_0}(f)(|0^{n+7}\rangle\,|0\rangle \otimes I_n) \tag{4.18}$$

and $\widetilde{P}_0(f)$ approximates $P_0(f)$ by the following inequalities:

$$\begin{aligned}\|\widetilde{P}_0(f) - P_0(f)\| &= \left\|(\langle 0| \otimes I_n)\left((\langle 0^{n+7}| \otimes I_1 \otimes I_n)U_{P_0}(f)(|0^{n+7}\rangle \otimes I_1 \otimes I_n) - \bar{P}_0(f)\right)(|0\rangle \otimes I_n)\right\| \\ &\le \left\|(\langle 0^{n+7}| \otimes I_1 \otimes I_n)U_{P_0}(f)(|0^{n+7}\rangle \otimes I_1 \otimes I_n) - \bar{P}_0(f)\right\| \\ &\le \epsilon_P.\end{aligned}$$

Therefore $U_{P_0}(f)$ is an $(1, n+8, \epsilon_P)$-block-encoding of $P_0(f)$.

As discussed in Section 4.4, the eigenpath we want to follow is $\{|0\rangle\,|x(f)\rangle\}$. However the approximate projection using eigenstate filtering only allows us to approximately follow this eigenpath. We denote the approximate states by $|\widetilde{x}(f_j)\rangle \approx |x(f_j)\rangle$, and will take into account the error of this approximation in our analysis.

With the block-encoding of $P_0(f)$ we can describe the algorithm is as follows:

1. Given $0 < \epsilon < 1$ and $\kappa > 1$ as well as the oracles mentioned at the beginning of Section 4.4. Set $M = \lceil \frac{4\log^2(\kappa)}{(1-1/\kappa)^2} \rceil$, $\epsilon_P = \frac{1}{162M^2}$.
2. Prepare $|\widetilde{x}(0)\rangle = |b\rangle$. Let $j = 1$.
3. Apply the $(1, n+8, \epsilon_P)$-block-encoding $U_{P_0}(f_j)$ of $P_0(f_j)$, constructed using eigenstate filtering with a polynomial of sufficiently high degree constructed in Lemma 20, to $|0^{n+8}\rangle |\widetilde{x}(f_{j-1})\rangle$ to get $U_{P_0}(f_j)(|0^{n+8}\rangle |\widetilde{x}(f_{j-1})\rangle)$.
4. Measure the $n+8$ ancilla qubits.
   (a) If not all outputs are 0 then abort and return to Step 2.
   (b) If all outputs are 0, and further $j < M-1$, then let $|\widetilde{x}(f_j)\rangle$ be the state in the main register that has not been measured, let $j \leftarrow j+1$, and go to Step 3. If all outputs are 0 and $j = M-1$ then go to next step.
5. Apply the $(1, n+8, \epsilon/4)$-block-encoding $U_{P_0}(1)$ of $P_0(1)$ to $|0^{n+8}\rangle |\widetilde{x}(f_{M-1})\rangle$ to get $U_{P_0}(f_j)(|0^{n+8}\rangle |\widetilde{x}(f_{M-1})\rangle)$.
6. Measure the $n+8$ ancilla qubits.
   (a) If not all outputs are 0 then abort and return to Step 2.
   (b) If all outputs are 0, then output $|\widetilde{x}(1)\rangle$ in the main register.

Here $|\widetilde{x}(f_j)\rangle$ are defined recursively in Steps 3 and 4 in the algorithm, starting with $|\widetilde{x}(0)\rangle = |b\rangle$. We can write down the recursion more concisely:

$$|\widetilde{x}(f_j)\rangle = \frac{\widetilde{P}_0(f_j)\, |\widetilde{x}(f_{j-1})\rangle}{\|\widetilde{P}_0(f_j)\, |\widetilde{x}(f_{j-1})\rangle \|}. \tag{4.19}$$

Going from $|\widetilde{x}(f_{j-1})\rangle$ to $|\widetilde{x}(f_j)\rangle$ has a success probability $\|\widetilde{P}_0(f_j)\, |\widetilde{x}(f_{j-1})\rangle \|^2$. We will show in the next section as well as in Appendix 4.13 that the the final success probability, which is the product of the success probabilities of these individual steps, does not go to 0. We emphasize that $\{|\widetilde{x}(f)\rangle\}$ is defined only for $f = f_j$ rather than arbitrary $f \in [0,1]$. We use this notation only to be consistent with the notation $|x(f)\rangle$.

Remark 28 (Choice of precision parameters)**.** *There are two precision parameters involved in the above discussion: ϵ and ϵ_P. Here ϵ is the target accuracy specified as part of our task, while ϵ_P is a parameter that is chosen by the algorithm according to Step 1, and is used only to ensure that the success probability is lower bounded by a* *constant. Also note that in the last step with $j = M$ (Steps 5 and 6), we set the target accuracy to be $\epsilon/4$ instead of ϵ_P in the previous steps. In fact, the errors of eigenstate filtering for $j = 1, 2, \ldots, M-1$ do not directly contribute to the final error. Rather, they only directly affect the success probability. When the overlap $|\langle\widetilde{x}(f_{M-1})|x(1)\rangle|$ is lower bounded by a constant away from 0, as we will show in Lemma 33, the final error is entirely controlled by the accuracy of the final eigenstate filtering for $j = M$, which is in turn controlled by the parameter $\epsilon/4$. In this way we ensure, as will be shown in the next section, that the output $|\widetilde{x}(1)\rangle$ satisfies*

$$|\langle\widetilde{x}(1)|x\rangle| \geq 1 - \epsilon.$$

Success probability, fidelity, and complexities

In this section we discuss the success probability of the algorithm described in the previous section, prove the fidelity of the output state is lower bounded by $1 - \epsilon$ for the given ϵ when ϵ_P and M are chosen as in Step 1 of the algorithm, and finally estimate the query and gate complexities.

We first give a lower bound for success probability assuming for simplicity each projection is done without error, i.e. $\epsilon_P = 0$. This is done so that we do not need to distinguish between eigenstates and approximate eigenstates produced using eigenstate filtering, thus making the derivation less technical. A rigorous lower bound, assuming a finite $\epsilon_P > 0$, will be given in Appendix 4.13. Under this assumption we have

$$P_{\text{succ}} = \prod_{j=1}^{M} \|P_0(f_j)\,|x(f_{j-1})\rangle\,\|^2 = \prod_{j=1}^{M} |\langle x(f_j)|x(f_{j-1})\rangle|^2.$$

Since

$$|\langle x(f_j)|x(f_{j-1})\rangle| \geq 1 - \frac{1}{2}\|\,|x(f_{j-1})\rangle - |x(f_j)\rangle\,\|^2, \tag{4.20}$$

$$\|\,|x(f_{j-1})\rangle - |x(f_j)\rangle\,\| \leq L(f_{j-1}, f_j) \leq L_*(f_{j-1}, f_j), \tag{4.21}$$

we have

$$\begin{aligned}
P_{\text{succ}} &\geq \left(\prod_{j=1}^{M}\left(1-\frac{1}{2}\|\,|x(f_{j-1})\rangle - |x(f_j)\rangle\,\|^2\right)\right)^2 \\
&\geq \left(1-\frac{2\log^2(\kappa)}{M^2(1-1/\kappa)^2}\right)^{2M} \\
&\geq \left(1-\frac{2\log^2(\kappa)}{M(1-1/\kappa)^2}\right)^{2} \\
&\geq \frac{1}{4},
\end{aligned}$$

where the we have used Eq. (4.15). This inequality holds for $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2}$ as required in the previous section.

Therefore we have shown the success probability is lower bounded by 1/4. The success probability when taking into account errors in each approximate projection, or in other words when we choose $\epsilon_P = 1/162M^2$ according to our algorithm rather than setting it to 0, is still lower bounded by a constant, which is proved in Appendix 4.13.

We then analyze the fidelity and complexities of our algorithm. Here we no longer assume $\epsilon_P = 0$, and the following discussion is therefore rigorous. In Appendix 4.13 it is shown that

$$|\,\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle\,| \geq 1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P} \geq \frac{1}{2}, \quad j = 0, 1, \ldots, M-1,$$

for $\epsilon_P \leq 1/128$ and $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2} \geq 4$. Therefore $|\,\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle\,| \geq 1/2$, which allows us to bound the error as,

$$\begin{aligned}
|\,\langle x|\widetilde{x}(1)\rangle\,| &= |\,\langle \widetilde{x}(f_M)|x(f_M)\rangle\,| \\
&= \frac{|\,\langle \widetilde{x}(f_{M-1})|\widetilde{P}_0(f_M)|x(f_M)\rangle\,|}{\|\widetilde{P}_0(f_M)\,|\widetilde{x}(f_{M-1})\rangle\,\|} \\
&\geq \frac{|\,\langle \widetilde{x}(f_{M-1})|P_0(f_M)|x(f_M)\rangle\,| - \epsilon/4}{\|P_0(f_M)\,|\widetilde{x}(f_{M-1})\rangle\,\| + \epsilon/4} \\
&= \frac{|\,\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle\,| - \epsilon/4}{|\,\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle\,| + \epsilon/4} \\
&\geq 1 - \frac{\epsilon/2}{|\,\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle\,|} \\
&\geq 1 - \epsilon.
\end{aligned} \tag{4.22}$$

The derivation is similar to that of Eq. (4.34), and we have used the fact that $\|\widetilde{P}_0(f_M) - P_0(f_M)\| \leq \epsilon/4$ because in Step 5 our algorithm in the previous section sets the eigenstate filtering accuracy to be $\epsilon/4$ instead of ϵ_P. Therefore the state $|\widetilde{x}(1)\rangle$ prepared in this way has a fidelity at least $1-\epsilon$.

We then estimate the computational costs. At each j we need to apply an $(1, n+8, \epsilon_P)$-block-encoding $U_{P_0}(f_j)$ of $P_0(f_j)$ to $|\widetilde{x}(f_{j-1})\rangle$ obtained form the last step. From the analysis in Appendix 4.13 we need $\epsilon_P \leq 1/162M^2$. Therefore we need to apply $U_H(f_j)$ and its inverse $\mathcal{O}\left(\frac{1-f_j+df_j}{\Delta_*(f_j)}\log(\frac{1}{\epsilon_P})\right)$ times. In total for $j = 1, 2, \ldots, M-1$ the number of queries to $U_H(f)$ is of the order

$$\begin{aligned}\log\left(\frac{1}{\epsilon_P}\right)\sum_{j=1}^{M-1}\frac{1-f(s_j)+f(s_j)d}{1-f(s_j)+f(s_j)/\kappa} &\leq \log\left(\frac{1}{\epsilon_P}\right)M\int_0^1\frac{1-f(s)+f(s)d}{1-f(s)+f(s)/\kappa}\mathrm{d}s \\ &= \log\left(\frac{1}{\epsilon_P}\right)M\left(\frac{d\kappa-1}{\log(\kappa)}-\frac{d-1}{1-1/\kappa}\right),\end{aligned} \tag{4.23}$$

for a d-sparse matrix A and κ is the condition number of A. Then in the last step for $j = M$, which is Step 5 in the algorithm in Section 4.5, we need to achieve accuracy $\epsilon/4$ for the eigenstate filtering. Therefore we need to apply the block-encoding $U_{P_0}(1)$ with $\mathcal{O}(d\kappa\log(\frac{1}{\epsilon}))$ queries to $U_H(1)$. As $M = \mathcal{O}(\log^2(\kappa))$, adding the query complexity of the last step to (4.23), and using the fact $\epsilon_P = \mathcal{O}(1/M^2)$, gives us the total query complexity of a single run

$$\mathcal{O}\left(d\kappa\left(\log(\kappa)\log\log(\kappa) + \log(1/\epsilon)\right)\right). \tag{4.24}$$

Because the success probability is $\Omega(1)$, the procedure needs to be run for an expected $\mathcal{O}(1)$ times to be successful, and therefore the total complexity remains the same. Since $U_H(f)$ queries $O_{A,1}$, $O_{A,2}$, and O_B each $\mathcal{O}(1)$ times, Eq. (4.24) is also the query complexity to these oracles.

Because the only thing we need to do in this method to solve QLSP is to repeatedly use QSP to do projection, no additional qubits are involved for time-dependent Hamiltonian simulation as in the previous AQC-based method. The total number of qubits is therefore $\mathcal{O}(n)$. The number of additional primitive gates required can be estimated similarly to the number of queries, which scales as $\mathcal{O}\left(nd\kappa\left(\log(\kappa)\log\log(\kappa)+\log(\frac{1}{\epsilon})\right)\right)$.

For the case when A is indefinite, we use a different pair of H_0 and H_1 as discussed in Appendix 4.10. The generalization to non-Hermitian matrices is the same as for Theorem 26, and it can be found in Appendix 4.10 as well. All other procedures are almost exactly the same. We summarize the results in the following theorem:

Theorem 29. *A is a d-sparse matrix whose singular values are in $[1/\kappa, 1]$ and can be queried through oracles $O_{A,1}$ and $O_{A,2}$ in (4.2), and $|b\rangle$ is given by an oracle O_B. Then $|x\rangle \propto A^{-1}|b\rangle$ can be obtained with fidelity $1-\epsilon$, succeeding with probability $\Omega(1)$ with ancilla qubits measurement outcome indicating success, using*

1. *$\mathcal{O}\left(d\kappa\left(\log(\kappa)\log\log(\kappa)+\log(\frac{1}{\epsilon})\right)\right)$ queries to $O_{A,1}$, $O_{A,2}$, and O_B,*
2. *$\mathcal{O}\left(nd\kappa\left(\log(\kappa)\log\log(\kappa)+\log(\frac{1}{\epsilon})\right)\right)$ other primitive gates,*
3. *$\mathcal{O}(n)$ qubits.*

The reason we put requirement on the singular values of A instead of its eigenvalues is stated in Remark 25. Just like in the case of AQC-based QLSP algorithm, here if we have $\mathcal{O}(\mathrm{poly}(n))$ gate complexity for the oracles $O_{A,1}$, $O_{A,2}$, and O_B, then the total gate complexity will be $\widetilde{\mathcal{O}}(\mathrm{poly}(n)d\kappa\log(1/\epsilon))$. Although we use $\mathcal{O}(n)$ qubits in total, the extra ancilla qubits we introduce in this method is in fact only $\mathcal{O}(1)$. This is a further improvement from the $\mathcal{O}(\log(d\kappa))$ ancilla qubits in the AQC-based QLSP algorithm.

We remark that there is the possibility to further slightly improve by a $\log(\kappa)$ factor (ignoring $\log\log$ terms) the asymptotic complexity of our QZE-based QLSP solver by using the fixed-point amplitude amplification to go from $|x(f_j)\rangle$ to $|x(f_{j+1})\rangle$ for each j, as discussed in [171, Corollary 1]. The bounds in this paper for many constant factors involved, particular those used in estimating the success probability of the QZE-based QLSP solver, are rather loose. However this does not concern us very much because we care mainly about the asymptotic complexity. Tighter estimates can be helpful for the actual implementation of our methods.

4.6 Discussion

In this paper, we have developed a quantum eigenstate filtering algorithm based on quantum signal processing (QSP). Our algorithm achieves the optimal query complexity among all polynomial-based eigenstate filtering methods, and uses a minimal amount of ancilla qubits. We demonstrate the usage of the eigenstate filtering method to solve quantum linear system problems (QLSP) with near-optimal complexity with respect to both the condition number κ and the accuracy ϵ. In the case when the precise value of κ is not known *a priori*, the knowledge of an upper bound of κ would suffice.

The problem of directly targeting at the solution $A^{-1}|b\rangle$ is that a (β, m, ϵ) block-encoding of A^{-1} requires at least $\beta \geq \kappa$ to make sure that $\|A^{-1}/\beta\| \leq 1$. Therefore the probability of success in the worst case is already $\Omega(\kappa^{-2})$, and the number of rounds of amplitude amplification needed is already $\mathcal{O}(\kappa)$. Therefore to achieve near-optimal complexity, this approach can only query the block-encoding of A for $\mathcal{O}(\text{polylog}(\kappa))$ times. To our best knowledge, there is no known method to achieve this for general matrices. However this might be possible for matrices with special structures and will be studied in future work.

Motivated by the success of AQC, our algorithm views QLSP as an eigenvalue problem, which can be implemented via $P\left|\widetilde{x}_0\right\rangle$, where P is an approximate projection operator, and $P\left|\widetilde{x}_0\right\rangle$ encodes the solution $|x\rangle$. The advantage of such a filtering procedure is that P is a projector and $\|P\| = 1$. Hence its (β, m, ϵ) block-encoding only requires $\beta \sim \mathcal{O}(1)$. Therefore assuming $\mathcal{O}(1)$ overlap between $\left|\widetilde{x}_0\right\rangle$ and the solution vector, which can be satisfied by running the time-optimal AQC to constant precision, the probability of success of the filtering procedure is already $\Omega(1)$ without any amplitude amplification procedure. This accelerates the query complexity of the recently developed time-optimal AQC from $\widetilde{\mathcal{O}}(\kappa/\epsilon)$ to $\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$. The efficient gate-based implementation of AQC still requires a time-dependent Hamiltonian simulation procedure (shown in Appendix 4.9). We then demonstrate that the dependence on the time-dependent Hamiltonian simulation procedure can be removed, using an algorithm based on the quantum Zeno effect, and the complexity is $\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$. Both algorithms have constant probability of success, and can prepare the solution in terms of a pure state.

It is worth noting that the eigenstate filtering method developed in this paper works only for the case when the eigenvalue corresponding to the desired eigenstate is known exactly, which is satisfied in the eigenvalue formulation of QLSP. In order to implement the QSP-based eigenstate filtering procedure, one still needs to find the phase factors associated with the block encoding $\widetilde{U}$. For a given polynomial $R_\ell(\cdot, \Delta)$, the phase factors are obtained on a classical computer in time that is polynomial in the degree and the logarithm of precision [85, Theorems 3-5]. However, this procedure requires solution of all roots of a high degree polynomial, which can be unstable for the range of polynomials $\ell \sim 100$ considered here. The stability of such procedure has recently been improved by Haah [92], though the number of bits of precision needed still scales as $\mathcal{O}(\ell \log(\ell/\epsilon))$. Significant progress has been achieved recently, enabling robust computation of phase factors for polynomials of degrees ranging from thousands to tens of thousands [51, 70]. We note that these phase factors in the eigenvalue filtering procedure only depend on $\widetilde{\Delta}$ and ℓ, and therefore can be reused for different matrices once they are obtained on a classical computer.

4.7 Block-encoding

The technique of block-encoding has been recently discussed extensively [86, 114]. Here we discuss how to construct block-encoding for $H - \lambda I$ which is used in eigenstate filtering, and Q_b, H_0, and H_1 which are used in QLSP and in particular the Hamiltonian simulation of AQC. We first introduce a simple technique we need to use repeatedly.

Given U_A, an $(\alpha, m, 0)$-block-encoding of A where $\alpha > 0$, we want to construct a block encoding of $A + cI$ for some $c \in \mathbb{C}$. This is in fact a special case of the linear combination of unitaries (LCU) technique introduced in [59]. Let

$$Q = \frac{1}{\sqrt{\alpha + |c|}} \begin{pmatrix} \sqrt{|c|} & -\sqrt{\alpha} \\ \sqrt{\alpha} & \sqrt{|c|} \end{pmatrix}$$

and $|q\rangle = Q|0\rangle$. Since $(\langle 0^m| \otimes I)U_A(|0^m\rangle \otimes I) = A/\alpha$, we have

$$(\langle q| \langle 0^m| \otimes I)(|0\rangle\langle 0| \otimes e^{i\theta} I + |1\rangle\langle 1| \otimes U_A)(|q\rangle |0^m\rangle \otimes I) = \frac{1}{\alpha + |c|}(A + cI),$$

where $\theta = \arg(c)$. Therefore Fig. 4.3 gives an $(\alpha + |c|, m + 1, 0)$-block-encoding of $e^{-i\theta}(A + cI)$.

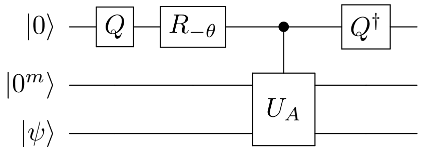


Figure 4.3: Quantum circuit for block-encoding of $e^{-i\theta}(A + cI)$, where $c = e^{i\theta}|c|$. $R_{-\theta}$= $|0\rangle\langle 0| + e^{-i\theta}|1\rangle\langle 1|$ is a phase shift gate. The three registers are the ancilla qubit for Q and $|q\rangle$, the ancilla register of U_A, and the main register, respectively.

Therefore we may construct an $(\alpha + |\lambda|, m + 1, 0)$-block-encoding of $H - \lambda I$. We remark that since $\lambda \in \mathbb{R}$, we can replace the phase shift gate with a Pauli-Z gate when $\lambda > 0$. This is at the same time a $(1, m + 1, 0)$-block-encoding of $\widetilde{H} = (H - \lambda I)/(\alpha + |\lambda|)$.

Now we construct a block-encoding of $Q_b = I - |b\rangle\langle b|$ with $|b\rangle = O_B|0\rangle$. Let $S_0 = I - 2|0^n\rangle\langle 0^n|$ be the reflection operator about the hyperplane orthogonal to $|0^n\rangle$. Then $S_b = O_B S_0 O_B^\dagger = I - 2|b\rangle\langle b|$ is the reflection about the hyperplane orthogonal to $|b\rangle$. Note that $Q_b = (S_b + I)/2$. Therefore we can use the technique illustrated in Fig. 4.3 to construct a $(1, 1, 0)$-block-encoding of Q_b. Here $|q\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Since $H_0 = \sigma_x \otimes Q_b$, we naturally obtain a $(1, 1, 0)$-block-encoding of H_0. We denote the block-encoding as U_{H_0}.

For the block-encoding of H_1, first note that

$$H_1 = \begin{pmatrix} I & 0 \\ 0 & Q_b \end{pmatrix} \begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & Q_b \end{pmatrix}.$$

From the block-encoding of Q_b, we can construct the block-encoding of controlled-Q_b by replacing all gates with their controlled counterparts. The block matrix in the middle is $\sigma_x \otimes A$. For a d-sparse matrix A, we have a $(d, n + 2, 0)$-block-encoding of A, and therefore we obtain a $(d, n + 2, 0)$-block-encoding of $\sigma_x \otimes A$. Then we can use the result for the product of block-encoded matrix [86, Lemma 30] to obtain a $(d, n + 4, 0)$-block-encoding of H_1, denoted by U_{H_1}.

The block-encodings of H_0 and H_1 allow us to block-encode linear combinations of them as well. We need access to $H(f) = (1-f)H_0 + fH_1$ which is used extensively in Section 4.5. This is done through [86, Lemma 29]. When applying the lemma we need the state preparation pair (P_L, P_R) such that

$$P_L |0\rangle = P_R |0\rangle = \frac{1}{\sqrt{1 - f + fd}}(\sqrt{1-f}\,|0\rangle + \sqrt{fd}\,|1\rangle).$$

The presence of the factor d is because H_1 is subnormalized by a factor of d in its block-encoding. By this lemma we obtain a $(1 - f + fd, n + 6, 0)$-block-encoding of $H(f)$. Here $1 - f + fd$ comes from the normalizing factor in the state preparation pair, and $n+6$ is the sum of the numbers of ancilla qubits used in the block-encodings of H_0 and H_1, plus one additional qubit used for the state preparation pair.

4.8 Implementing the reflection operator and θ-reflection operator

In this appendix we prove Theorem 22 and Corollary 23 by constructing the quantum circuits. In both the theorem and the corollary we assume, as in Theorem 21, that H is a Hermitian matrix and U_H is an $(\alpha, m, 0)$-block-encoding of H. Also λ is an eigenvalue of H that is separated from the rest of the spectrum by a gap Δ.

We first prove Theorem 22 by constructing the circuit for the reflection operator

$$R_\lambda = 2P_\lambda - I,$$

where P_λ is the projection operator into the λ-eigenspace of H. To do this we use the following polynomial

$$S_\ell(x;\delta) = \frac{2R_\ell(x;\delta)-1}{\max_{y\in[-1,1]}|2R_\ell(y;\delta)-1|}.$$

The first thing we should notice about this polynomial is that it is even and therefore can be implemented via QSP by Theorem 1'. The normalization is done so that we have $|S_\ell(x;\delta)| \leq 1$ for all $x \in [-1,1]$. Because $-\epsilon \leq \min_{y\in[-1,1]} R_\ell(y;\delta) < 0$ and $\max_{y\in[-1,1]} R_\ell(y;\delta) = 1$, we have

$$1 \leq \max_{y\in[-1,1]} |2R_\ell(y;\delta)-1| \leq 1+2\epsilon.$$

Therefore

$$-1-2\epsilon \leq S_\ell(x;\delta) \leq \frac{-1+2\epsilon}{1+2\epsilon} \leq -1+4\epsilon, \quad x \in \mathcal{D}_\delta, \tag{4.25}$$

and

$$1-2\epsilon \leq \frac{1}{1+2\epsilon} \leq S_\ell(0;\delta) \leq 1. \tag{4.26}$$

Now for H, we define $\widetilde{H} = (H-\lambda I)/(\alpha+|\lambda|)$ and $\widetilde{\Delta} = \Delta/2\alpha$ as done in the proof of of Theorem 21. Then applying the polynomial $S_\ell(x;\widetilde{\Delta})$ to $\widetilde{H}$, because all eigenvalues of $\widetilde{H}$ are contained in $\mathcal{D}_{\widetilde{\Delta}} \cup \{0\}$, they are mapped to either close to 1 or close to -1. Thus by Eqs. (4.25) and (4.26) we have

$$\|S_\ell(\widetilde{H};\widetilde{\Delta}) - R_\lambda\| \leq 4\epsilon.$$

Since $S_\ell(x;\widetilde{\Delta})$ is a real even polynomial that takes value in $[-1,1]$ when $x \in [-1,1]$, we can implement a $(1,m+2,0)$-block-encoding of $S_\ell(\widetilde{H};\widetilde{\Delta})$ through QSP by Theorem 1'. We denote this block-encoding by $\mathcal{U}_R$. We have

$$\|(\langle 0^{m+2}| \otimes I)\mathcal{U}_R(|0^{m+2}\rangle \otimes I) - R_\lambda\| = \|S_\ell(\widetilde{H};\widetilde{\Delta}) - R_\lambda\| \leq 4\epsilon.$$

Therefore $\mathcal{U}_R$ is an $(1,m+2,4\epsilon)$-block-encoding of R_λ. Thus we have proved Theorem 22.

We then prove Corollary 23 by constructing a block-encoding of the θ-reflection operator

$$P_\lambda + e^{\mathrm{i}\theta}(I-P_\lambda).$$

One might be tempted to directly find a polynomial to approximate this matrix function. However such a polynomial would have complex coefficients, and we would need to apply QSP to the real and imaginary parts separately. This in turn needs an extra LCU step to add the two parts up, resulting in reduced success probability. Therefore instead of using a new polynomial, we use the block-encoding $\mathcal{U}_R$ we have already constructed, and then apply a 1-bit phase estimation on it. This enables u s to distinguish between the λ-eigenspace and its orthogonal complement, since all the eigenvalues of R_λ are either 1 or -1. We then apply the phase factor $e^{\mathrm{i}\theta}$ only to the correct subspace. Finally we uncompute the additional ancilla qubit. The circuit takes the following form, as shown in Figure 4.4:

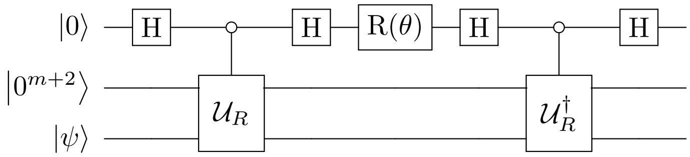


Figure 4.4: The quantum circuit for implementing the θ-reflection operator. H is the Hadamard gate and $\mathrm{R}(\theta) = |0\rangle\langle 0| + e^{\mathrm{i}\theta}|1\rangle\langle 1|$ is the phase-shift gate.

We introduced one additional ancilla qubit in the initial state $|0\rangle$, and the second register in the above circuit is for the ancilla qubits in Theorem 22. The last register is the main register prepared in the state $|\psi\rangle$ on which we want to apply the operator $P_\lambda + e^{\mathrm{i}\theta}(I - P_\lambda)$. Thus we have proved Corollary 23.

4.9 Gate-based implementation of time-optimal adiabatic quantum computing

In Theorem 26 we used an adiabatic time evolution to prepare an initial state for eigenstate filtering. In this appendix we discuss how to implement this time evolution on a gate-based quantum computer. Consider the adiabatic evolution

$$\frac{1}{T}\mathrm{i}\partial_s |\psi_T(s)\rangle = H(f(s))\,|\psi_T(s)\rangle\,, \quad |\psi_T(0)\rangle = |0\rangle\,|b\rangle\,,$$

Where $H(f) = (1-f)H_0 + fH_1$ for H_0 and H_1 defined in (4.5) and (4.4). It is proved in [15, 154] that the gap between 0 and the rest of the eigenvalues of $H(f)$ is lower bounded by $1 - f + f/\kappa$. With this bound the scheduling (4.12) in the AQC(p) scheme results in $\mathcal{O}(\kappa/\epsilon)$ runtime complexity to solve QLSP. As mentioned before, the fact that the 0-eigenspace of $H(f(s))$ is two dimensional is not a problem because $|\psi_T(t)\rangle$ is orthogonal to $|1\rangle|b\rangle$ for all t.

In order to carry out AQC efficiently using a gate-based implementation, we use the recently developed time-dependent Hamiltonian simulation method based on truncated Dyson series introduced in [116]. In Hamiltonian simulation, several types of input models for the Hamiltonian are in use. Hamiltonians can be input as a linear combination of unitaries [35], using its sparsity structure [5, 115], or using its block-encoding [114, 116]. For a time-dependent Hamiltonian Low and Wiebe designed an input model based on block-encoding named HAM-T [116, Definition 2], as a block-encoding of $\sum_s |s\rangle\langle s| \otimes H(s)$ where s is a time step and $H(s)$ is the Hamiltonian at this time step.

In the gate-based implementation of the time-optimal AQC, we construct HAM-T in Fig. 4.5. We need to use the block-encodings U_{H_0} and U_{H_1} introduced in Appendix 4.7, which requires $n_0 = 1$ and $n_1 = n+4$ ancilla qubits, respectively. Our construction of HAM-T satisfies

$$(\langle s|\langle 0^{l+1+n_0}| \otimes I \otimes \langle 0^{n_1+1}|)\text{HAM-T}(|s\rangle|0^{l+1+n_0}\rangle \otimes I \otimes |0^{n_1+1}\rangle) = H(f(s))/d, \quad (4.27)$$

for any $s \in \mathcal{S} = \{j/2^l : j = 0, 1, \ldots, 2^l - 1\}$.

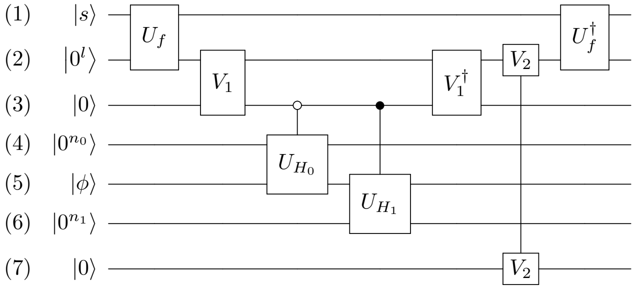


Figure 4.5: Quantum circuit for HAM-T. The registers from top to bottom are: (1) input register for s (2) register for storing $f(s)$ (3) register for a control qubit (4) ancilla register for U_{H_0} (5) main register for input state $|\phi\rangle$ (6) ancilla register for U_{H_1} (7) register for changing normalizing factor from $\alpha(s)$ to d.

In this unitary HAM-T we also need the unitary

$$U_f|s\rangle|z\rangle = |s\rangle|z \oplus f(s)\rangle \quad (4.28)$$

to compute the scheduling function needed in the time-optimal AQC, and the unitaries

$$\begin{aligned} V_1 &= \sum_{s\in\mathcal{S}} |s\rangle\langle s| \otimes \frac{1}{\sqrt{1-s+ds}} \begin{pmatrix} \sqrt{1-s} & -\sqrt{ds} \\ \sqrt{ds} & \sqrt{1-s} \end{pmatrix} \\ V_2 &= \sum_{s\in\mathcal{S}} |s\rangle\langle s| \otimes \begin{pmatrix} \frac{\alpha(s)}{d} & -\sqrt{1-\left(\frac{\alpha(s)}{d}\right)^2} \\ \sqrt{1-\left(\frac{\alpha(s)}{d}\right)^2} & \frac{\alpha(s)}{d} \end{pmatrix}, \end{aligned} \tag{4.29}$$

where $\alpha(s) = 1 - s + ds$. Here V_1 is used for preparing the linear combination $(1-f(s))U_{H_0} + f(s)U_{H_1}$. Without V_2 the circuit would be a $(\alpha(s), l+n_0+n_1+2, 0)$-block-encoding of $\sum_s |s\rangle\langle s| \otimes H(s)$, but with V_2 it becomes a $(d, l+n_0+n_1+2, 0)$-block-encoding, so that the normalizing factor is time-independent, as is required for the input model in [116].

For the AQC with positive definite A we have $n_0 = 1$ and $n_1 = n+4$. For the Hermitian indefinite case we have $n_0 = 2$ and $n_1 = n+4$. The increase of n_0 from 1 to 2 is due to the additional operation of linear combination of matrices. For H_1 we can perform one less matrix-matrix multiplication, and hence the value of n_1 remains unchanged (see Appendix 4.10).

Following [116, Corollary 4], we may analyze the different components of costs in the Hamiltonian simulation of AQC. For time evolution from $s = 0$ to $s = 1$, HAM-T is a $(dT, l+n_0+n_1+2, 0)$-block-encoding of $\sum_s |s\rangle\langle s| \otimes TH(s)$. With the scheduling function given in [15] we have $\|TH(s)\| = \mathcal{O}(Td)$ and $\|\frac{\mathrm{d}(TH(s))}{\mathrm{d}s}\| = \mathcal{O}(dT\kappa^{p-1})$. We choose $p = 1.5$ and by [15, Theorem 1] we have $T = \mathcal{O}(\kappa)$. We only need to simulate up to constant precision, and therefore we can set $l = \mathcal{O}(\log(d\kappa))$. The costs are then

1. Queries to HAM-T: $\mathcal{O}\left(d\kappa\frac{\log(d\kappa)}{\log\log(d\kappa)}\right)$,
2. Primitive gates: $\mathcal{O}\left(d\kappa(n+\log(d\kappa))\frac{\log(d\kappa)}{\log\log(d\kappa)}\right)$,
3. Qubits: $\mathcal{O}(n+\log(d\kappa))$.

4.10 The matrix dilation method

In Theorem 26 and Theorem 29, in order to extend the time-optimal AQC method, and the QZE-based method to Hermitian indefinite matrices, we follow [15, Theorem

2], where H_0 and H_1, as constructed in Ref. [154] are given by

$$\begin{aligned} H_0 &= \sigma_+ \otimes [(\sigma_z \otimes I_N)Q_{+,b}] + \sigma_- \otimes [Q_{+,b}(\sigma_z \otimes I_N)], \\ H_1 &= \sigma_+ \otimes [(\sigma_x \otimes A)Q_{+,b}] + \sigma_- \otimes [Q_{+,b}(\sigma_x \otimes A)]. \end{aligned} \tag{4.30}$$

Here $\sigma_\pm = (\sigma_x \pm i\sigma_y)/2$ and $Q_{+,b} = I_{2N} - \left|+\right\rangle\left|b\right\rangle\left\langle+\right|\left\langle b\right|$. The dimension of the dilated matrices H_0, H_1 is $4N$. The lower bound for the gap of $H(f)$ then becomes $\sqrt{(1-f)^2 + f^2/\kappa^2}$ [154]. However in order to simplify our analysis we give a weaker lower bound

$$\Delta_*(f) = \frac{1}{\sqrt{2}}\left(1 - f + \frac{f}{\kappa}\right),$$

which differs from the gap lower bound in (4.6) by a factor of $\sqrt{2}$. The initial state is $\left|0\right\rangle\left|-\right\rangle\left|b\right\rangle$, where $\left|-\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$, and the goal is to obtain $\left|0\right\rangle\left|+\right\rangle\left|x\right\rangle$. In the AQC-based QLSP solver, after running the AQC we can remove the second qubit by measuring it with respect to the $\{\left|+\right\rangle, \left|-\right\rangle\}$ basis and accepting the result corresponding to $\left|+\right\rangle$. The resulting query complexity remains unchanged. We remark that the matrix dilation here is only needed for AQC. The eigenstate filtering procedure can still be applied to the original matrix of dimension $2N$. The same is true for the QZE-based method.

For a general matrix, we may first consider the extended linear system. Define an extended QLSP $\mathfrak{A}\left|\mathfrak{x}\right\rangle = \left|\mathfrak{b}\right\rangle$ in dimension $2N$ where

$$\mathfrak{A} = \sigma_+ \otimes A + \sigma_- \otimes A^\dagger = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}, \quad \left|\mathfrak{b}\right\rangle = \left|0, b\right\rangle.$$

Here $\mathfrak{A}$ is a Hermitian matrix of dimension $2N$, with condition number κ and $\|\mathfrak{A}\| = 1$, and $\left|\mathfrak{x}\right\rangle = \left|1, x\right\rangle$ solves the extended QLSP. Therefore the time-optimal AQC and the QZE procedure can be applied to the Hermitian matrix $\mathfrak{A}$ to prepare an ϵ-approximation of x. The dimension of the corresponding H_0, H_1 matrices is $8N$. Again the matrix dilation method used in Eq. (4.30) is not needed for the eigenstate filtering step.

4.11 Optimality of the Chebyshev filtering polynomial

In this section we prove Lemma 20. We define

$$Q_\ell(x; \Delta) = T_\ell\left(-1 + 2\frac{x^2 - \Delta^2}{1 - \Delta^2}\right),$$

then $R_\ell(x;\Delta) = Q_\ell(x;\Delta)/Q_\ell(0;\Delta)$. Here $T_\ell(x)$ is the ℓ-th Chebyshev polynomial of the first kind and $0 < \Delta < 1$. We need to use the following lemma, which is similar to the well-known result discussed in [144, Proposition 2.4], [143, Theorem 6.25], and [74, Theorem 7]:

Lemma 30. *For any $p(x) \in \mathbb{P}_{2\ell}[x]$ satisfying $|p(x)| \leq 1$ for all $x \in \mathcal{D}_\Delta$, where $\mathcal{D}_\Delta = [-1,-\Delta] \cup [\Delta, 1]$, $|Q_\ell(x;\Delta)| \geq |p(x)|$ for all $x \notin \mathcal{D}_\Delta$.*

Proof. We prove by contradiction. If there exists $q(x) \in \mathbb{P}_{2\ell}[x]$ such that $|q(x)| \leq 1$ for all $x \in \mathcal{D}_\Delta$ and there exists $y \notin \mathcal{D}_\Delta$ such that $|q(y)| > |Q_\ell(y;\Delta)|$, then letting $h(x) = Q_\ell(x;\Delta) - q(x)\frac{Q_\ell(y;\Delta)}{q(y)}$, we want to show $h(x)$ has at least $2\ell+1$ distinct zeros.

First note that there exist $-1 = y_1 < y_2 < \cdots < y_{\ell+1} = 1$ such that $|T_\ell(y_j)| = 1$, and $T_\ell(y_j)T_\ell(y_{j+1}) = -1$. Therefore there exist $\Delta = x_1 < x_2 < \cdots < x_{\ell+1} = 1$ such that $|Q_\ell(\pm x_j;\Delta)| = 1$, and $Q_\ell(x_j;\Delta)Q_\ell(x_{j+1};\Delta) = -1$. In other words, $Q_\ell(\cdot;\Delta)$ maps each (x_j, x_{j+1}) and $(-x_{j+1}, -x_j)$ to $(-1, 1)$, and the mapping is bijective for each interval. Because $|\frac{Q_\ell(y;\Delta)}{q(y)}| < 1$, there exists $z_j, w_j \in (x_j, x_{j+1})$ for each j such that $h(z_j) = h(-w_j) = 0$. Therefore $\{z_j\}$ and $\{-w_j\}$ give us 2ℓ distinct zeros. Another zero can be found at y as $h(y) = Q_\ell(y) - Q_\ell(y) = 0$. Therefore there are $2\ell + 1$ distinct zeros.

However $h(x)$ is of degree at most 2ℓ. This shows $h(x) \equiv 0$. This is clearly impossible since $h(1) = Q_\ell(1;\Delta) - q(1)\frac{Q_\ell(y;\Delta)}{q(y)} = 1 - q(1)\frac{Q_\ell(y;\delta)}{q(y)} > 0$. □

Lemma 30 shows that for any $y \notin \mathcal{D}_\Delta$,

$$\max_{\substack{p(x)\in\mathbb{P}_{2\ell}[x] \\ |p(x)|\leq 1, \forall x\in\mathcal{D}_\Delta}} |p(y)| = |Q_\ell(y;\Delta)|.$$

This is equivalent to

$$\max_{p(x)\in\mathbb{P}_{2\ell}[x]} \frac{|p(y)|}{\max_{x\in\mathcal{D}_\Delta}|p(x)|} = |Q_\ell(y;\Delta)|,$$

which is in turn equivalent to

$$\min_{p(x)\in\mathbb{P}_{2\ell}[x]} \frac{\max_{x\in\mathcal{D}_\Delta}|p(x)|}{|p(y)|} = \frac{1}{|Q_\ell(y;\Delta)|},$$

and

$$\min_{\substack{p(x)\in\mathbb{P}_{2\ell}[x] \\ |p(y)|\leq 1}} \max_{x\in\mathcal{D}_\Delta} |p(x)| = \frac{1}{|Q_\ell(y;\Delta)|}.$$

This implies (i) of Lemma 20: we only need to set $y = 0$ and observe that

$$\max_{x\in\mathcal{D}_\Delta} |R_\ell(x;\Delta)| = \frac{1}{|Q_\ell(0;\Delta)|},$$

since the Chebyshev polynomials take value between $[-1, 1]$ on the interval $[-1, 1]$. From the above discussion we may derive a more general result, that $R_\ell(x;\Delta)$ solves the following minimax problem:

$$\underset{\substack{p(x)\in\mathbb{P}_{2\ell}[x]\\ p(y)=R_\ell(y;\Delta)}}{\text{minimize}} \max_{x\in\mathcal{D}_\Delta} |p(x)|.$$

To prove (ii) of Lemma 20, we need to use the following lemma, which directly follows from [143, Eq. (6.112)]:

Lemma 31. *Let $T_\ell(x)$ be the ℓ-th Chebyshev polynomial, then*

$$T_\ell(1+\delta) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$$

for $0 \leq \delta \leq 3 - 2\sqrt{2}$.

Proof. The Chebyshev polynomial can be rewritten as $T_\ell(x) = \frac{1}{2}(z^\ell + \frac{1}{z^\ell})$ for $x = \frac{1}{2}(z + \frac{1}{z})$. Let $x = 1 + \delta$, then $z = 1 + \delta \pm \sqrt{2\delta + \delta^2}$. The choice of ± does not change the value of x, so we choose $z = 1 + \delta + \sqrt{2\delta + \delta^2} \geq 1 + \sqrt{2\delta}$. Since $\log(1 + \sqrt{2\delta}) \geq \sqrt{2\delta} - \delta \geq \sqrt{\delta}$ for $0 \leq \delta \leq 3 - 2\sqrt{2}$, we have $z^\ell \geq e^{\ell\sqrt{\delta}}$. Thus $T_\ell(x) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$. □

We use this lemma to prove (ii). Since $|T_\ell(-1 + 2\frac{-\Delta^2}{1-\Delta^2})| \geq T_\ell(1 + 2\Delta^2)$, when $\Delta^2 \leq 1/12$, we have $2\Delta^2 \leq 1/6 < 3 - 2\sqrt{2}$. Thus by the above lemma we have $|T_\ell(-1 + 2\frac{-\Delta^2}{1-\Delta^2})| \geq \frac{1}{2}e^{\ell\sqrt{2\Delta^2}}$. Since $|T_\ell(-1 + 2\frac{x^2-\Delta^2}{1-\Delta^2})| \leq 1$ for $x \in \mathcal{D}_\Delta$, we have the inequality in (ii). (iii) follows straightforwardly from the monotonicity of Chebyshev polynomials outside of $[-1, 1]$.

4.12 Properties of the eigenpath

In this section we construct a smooth one-parameter family of normalized quantum states $\{|x(f)\rangle\}$ satisfying Eqs. (4.7) and (4.8). $\{|0\rangle\,|x(f)\rangle\}$ then gives an eigenpath of the one-parameter family of Hamiltonians $\{H(f)\}$. We also prove the inequality (4.9).

We define

$$|y(f)\rangle = ((1-f)I + fA)^{-1}|b\rangle .$$

Then $\langle y(f)|y(f)\rangle \geq 1$ because $\|(1-f)I+fA\| \leq 1$. Also $|y(f)\rangle$ is a smooth function of f for $f \in (0,1)$ because $(1-f)I + fA$ is invertible in this interval, under the assumption that A is Hermitian positive-definite with eigenvalues in $[1/\kappa, 1]$. We construct $|x(f)\rangle$ through

$$|x(f)\rangle = c(f)\,|y(f)\rangle ,$$

with $c(f)$ solving the following ODE

$$c'(f) = -c(f)\frac{\langle y(f)|\partial_f|y(f)\rangle}{\langle y(f)|y(f)\rangle}, \quad c(0) = 1. \tag{4.31}$$

This is a linear ODE and the right-hand side depends smoothly on f. Therefore the solution exists and is unique for $f \in [0,1]$. It then follows that this construction of $|x(f)\rangle$ satisfies (4.7). Since

$$\partial_f |x(f)\rangle = c'(f)\,|y(f)\rangle + c(f)\partial_f\,|y(f)\rangle ,$$

we have

$$\langle x(f)|\,\partial_f\,|x(f)\rangle = c^*(f)[c'(f)\,\langle y(f)|y(f)\rangle + c(f)\,\langle y(f)|\,\partial_f\,|y(f)\rangle] = 0.$$

Therefore Eq. (4.8) is satisfied, and this in turn ensures $|x(f)\rangle$ is normalized. In this way we have constructed $\{|x(f)\rangle\}$ that satisfies all the requirements in Section 4.4. For $H(f) = (1-f)H_0 + fH_1$, where H_0 and H_1 are defined in Eqs. (4.5) and (4.4) respectively, we can see $H(f)\,|\bar{x}(f)\rangle = 0$ where $|\bar{x}(f)\rangle = |0\rangle\,|x(f)\rangle$. Therefore $\{|\bar{x}(f)\rangle\}$ is a smooth eigenpath.

If there is another eigenpath $\{|0\rangle\,|w(f)\rangle\}$ satisfying $\langle w(f)|\partial_f|w(f)\rangle = 0$, then it follows that $((1-f)I + fA)\,|w(f)\rangle \propto |b\rangle$. Therefore $|w(f)\rangle = e^{\mathrm{i}\theta(f)}\,|x(f)\rangle$ for some differentiable $\theta(f)$. By the geometric phase condition we can show $e^{\mathrm{i}\theta(f)} = 1$ for all f by also taking into account the initial condition $|w(f)\rangle = |b\rangle$, and therefore $|w(f)\rangle = |x(f)\rangle$. This proves uniqueness.

Now we denote by $\varepsilon_j(f)$ the eigenvalues of $H(f)$. The corresponding eigenstates are denoted by $|w_j(f)\rangle$. Because $((1-f)I+fA)\,|x(f)\rangle \propto |b\rangle$, we have $H(f)\,|\bar{x}(f)\rangle = 0$. Since $|x(f)\rangle$, and as a result $|\bar{x}(f)\rangle$, is differentiable, taking derivative with respect to f we have

$$H'(f)\,|\bar{x}(f)\rangle + H(f)\partial_f\,|\bar{x}(f)\rangle = 0.$$

Therefore

$$\langle w_j(f)|\,H'(f)\,|\bar{x}(f)\rangle + \langle w_j(f)|\,H(f)\partial_f\,|\bar{x}(f)\rangle = 0.$$

And this leads to

$$\langle w_j(f)| \partial_f |\bar{x}(f)\rangle = -\frac{\langle w_j(f)| H'(f) |\bar{x}(f)\rangle}{\varepsilon_j(f)}$$

for any j such that $\varepsilon_j(f) \neq 0$. The null space of $H(f)$ is spanned by $|1\rangle |b\rangle$ and $|\bar{x}(f)\rangle$. We have $(\langle 1| \langle b|) |\bar{x}(f)\rangle = \langle 1|0\rangle \langle b|x(f)\rangle = 0$, and $\langle \bar{x}(f)|\partial_f|\bar{x}(f)\rangle = 0$ because of the geometric phase condition (4.8). Since all $|w_j(f)\rangle$ such that $\varepsilon_j(f) \neq 0$, together with $|1\rangle |b\rangle$ and $|\bar{x}(f)\rangle$ form a basis of the Hilbert space, we have

$$\partial_f |\bar{x}(f)\rangle = -\sum_{j:\varepsilon_j(f)\neq 0} \frac{|w_j(f)\rangle \langle w_j(f)| H'(f) |\bar{x}(f)\rangle}{\varepsilon_j(f)}$$

Therefore

$$\begin{aligned}\|\partial_f |\bar{x}(f)\rangle \|^2 &= \sum_{j:\varepsilon_j(f)\neq 0} \frac{|\langle w_j(f)| H'(f) |\bar{x}(f)\rangle|^2}{\varepsilon_j^2(f)} \\ &\leq \frac{1}{\Delta_*(f)^2} \sum_{j:\varepsilon_j(f)\neq 0} |\langle w_j(f)| H'(f) |\bar{x}(f)\rangle|^2 \\ &\leq \frac{1}{\Delta_*(f)^2} \|H'(f) |\bar{x}(f)\rangle \|^2\end{aligned}$$

From the definition of $H(f)$ it can be seen that $\|H'(f)\| \leq 2$. Therefore we have proved the inequality (4.9).

4.13 Success probability of Quantum Zeno effect QLSP algorithm

In this appendix we rigorously prove a constant success probability lower bound for the QZE-based QLSP algorithm in Theorem 29. In Section 4.5 we gave a simpler but non-rigorous proof of a constant success probability lower bound by assuming the projection for each $H(f_j)$ is done without error, i.e. $\epsilon_P = 0$. Here we do not make such an assumption and show we can still find such a lower bound. We will need to use the following elementary inequality, which can be easily proved using induction.

Lemma 32. *If $0 < a_j < 1$, $0 < b_j < 1$, $j = 0, 1, 2, \ldots, R-1$, then*

$$\prod_{j=0}^{R-1} (a_j - b_j) \geq \prod_{j=0}^{R-1} a_j - \sum_{j=0}^{R-1} b_j.$$

We first recall the definition of the sequence of quantum states $\{|x(f_j)\rangle\}$, with each $|x(f_j)\rangle$ defined through (4.7) and (4.8), satisfying $H(f_j)\,|0\rangle\,|x(f_j)\rangle = 0$, and the sequence of quantum states $\{|\widetilde{x}(f_j)\rangle\}$, with each $|\widetilde{x}(f_j)\rangle$ defined recursively by (4.19). We need to use the following bound for the overlap between $|x(f_j)\rangle$ and $|x(f_{j+1})\rangle$ derived from Eqs. (4.20) (4.21) and (4.15).

$$|\langle x(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2}\|\,|x(f_{j+1})\rangle - |x(f_j)\rangle\,\|^2 \geq 1 - \frac{2\log^2(\kappa)}{M^2(1-1/\kappa)^2}. \tag{4.32}$$

With these tools we will first bound several overlaps in the following lemma

Lemma 33. *When $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2}$ and $\epsilon_P \leq \frac{1}{128}$, we have for $j = 0, 1, \dots, M-1$:*
(i) $|\langle x(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2M}$,
(ii) $|\langle x(f_j)|\widetilde{x}(f_j)\rangle| \geq 1 - 4\epsilon_P$,
(iii) $|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P}$.

Proof. (i) derives directly from (4.32). We then want to derive (ii) and (iii) inductively. First we have

$$\begin{aligned}|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| &= |\langle \widetilde{x}(f_j)|P_0(f_j)|x(f_{j+1})\rangle + \langle \widetilde{x}(f_j)|I - P_0(f_j)|x(f_{j+1})\rangle| \\ &\geq |\langle x(f_j)|x(f_{j+1})\rangle| \cdot |\langle x(f_j)|\widetilde{x}(f_j)\rangle| - \|(I - P_0(f_j))\,|\widetilde{x}(f_j)\rangle\|.\end{aligned}$$

Because

$$\|(I - P_0(f_j))\,|\widetilde{x}(f_j)\rangle\,\|^2 = 1 - |\langle x(f_j)|\widetilde{x}(f_j)\rangle|^2,$$

we then have

$$|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| \geq |\langle x(f_j)|x(f_{j+1})\rangle| \cdot |\langle x(f_j)|\widetilde{x}(f_j)\rangle| - \sqrt{1 - |\langle x(f_j)|\widetilde{x}(f_j)\rangle|^2}.$$

We denote

$$|\langle x(f_j)|\widetilde{x}(f_j)\rangle| = 1 - \nu_j,$$

then

$$\begin{aligned}|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| &\geq (1 - \frac{1}{2M})(1-\nu_j) - \sqrt{1-(1-\nu_j)^2} \\ &\geq 1 - \frac{1}{2M} - \nu_j - \sqrt{2\nu_j}.\end{aligned} \tag{4.33}$$

We now bound ν_{j+1} using $|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle|$. First using the fact that $\|\widetilde{P}_0(f_j) - P_0(f_j)\| \leq \epsilon_P$ where the approximate projection operator $\widetilde{P}_0(f_j)$ is defined in Eq. (4.18), we have

$$\begin{aligned}|\langle\widetilde{x}(f_{j+1})|x(f_{j+1})\rangle| &= \frac{|\langle\widetilde{x}(f_j)|\widetilde{P}_0(f_{j+1})|x(f_{j+1})\rangle|}{\|\widetilde{P}_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\|} \\ &\geq \frac{|\langle\widetilde{x}(f_j)|P_0(f_{j+1})|x(f_{j+1})\rangle| - \epsilon_P}{\|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| + \epsilon_P} \\ &= \frac{|\langle\widetilde{x}(f_j)|x(f_{j+1})\rangle| - \epsilon_P}{|\langle\widetilde{x}(f_j)|x(f_{j+1})\rangle| + \epsilon_P} \\ &\geq 1 - \frac{2\epsilon_P}{|\langle\widetilde{x}(f_j)|x(f_{j+1})\rangle|}.\end{aligned} \tag{4.34}$$

This leads to

$$\nu_{j+1} \leq \frac{2\epsilon_P}{|\langle\widetilde{x}(f_j)|x(f_{j+1})\rangle|} \leq \frac{2\epsilon_P}{1 - \frac{1}{2M} - \nu_j - \sqrt{2\nu_j}}, \tag{4.35}$$

which establishes a recurrence relation for ν_j. Because $\nu_0 = 0$, $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2} \geq 4$ and $\epsilon_P \leq \frac{1}{128}$, we can prove inductively that $\nu_j \leq \frac{1}{32}$. Taking this into (4.35) we have

$$\nu_{j+1} \leq 4\epsilon_P,$$

which proves (ii). Taking this into (4.33) we have (iii).

□

An immediate corollary of (iii) in the above lemma is

$$|\langle\widetilde{x}(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P} \geq \frac{1}{2}, \tag{4.36}$$

for $j = 0, 1, \ldots, M-1$, $M \geq 4$, and $\epsilon_P \leq 1/128$. With these tools we are now ready to estimate the success probability P_{succ}. We have

$$\begin{aligned}
P_{\text{succ}} &= \prod_{j=0}^{M-1} \|\widetilde{P}_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\|^2 \\
&\geq \left(\prod_{j=0}^{M-2} (\|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| - \epsilon_P)\right)^2 \left(\|P_0(1)\,|\widetilde{x}(f_{M-1})\rangle\,\| - \frac{\epsilon}{4}\right)^2 \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-2} (\|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| - \epsilon_P)\right)^2 \qquad (4.37) \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-1} (\|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| - \epsilon_P)\right)^2 \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-1} \|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| - M\epsilon_P\right)^2 .
\end{aligned}$$

In the last line we have used Lemma 32. In the second line the $j = M-1$ case is treated differently because in the last step we need to attain $\epsilon/4$ precision for eigenstate filtering. We bound the success probability of the last step using

$$\left(\|P_0(1)\,|\widetilde{x}(f_{M-1})\rangle\,\| - \frac{\epsilon}{4}\right)^2 = \left(\|\,\langle x(f_M)|\widetilde{x}(f_{M-1})\rangle\,\| - \frac{\epsilon}{4}\right)^2 \geq \left(\frac{1}{2} - \frac{1}{4}\right)^2 = \frac{1}{16},$$

where we have used Eq. (4.36) for $j = M-1$. This inequality motivates us to bound $\prod_{j=0}^{M-1} \|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\|$, for which, by Lemma 33, we have

$$\begin{aligned}
\prod_{j=0}^{M-1} \|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| &= \prod_{j=0}^{M-1} |\,\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle\,| \\
&\geq \left(1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P}\right)^M \qquad (4.38) \\
&\geq \frac{1}{2} - M(4\epsilon_P + 2\sqrt{2\epsilon_P}).
\end{aligned}$$

In Lemma 33 we have required that $\epsilon_P \leq \frac{1}{128}$ and $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2} \geq 4$. Therefore when we further require $\epsilon_P \leq \frac{1}{162M^2}$ we have

$$\prod_{j=0}^{M-1} \|P_0(f_{j+1})\,|\widetilde{x}(f_j)\rangle\,\| \geq \frac{1}{4}.$$

Substituting this into (4.37) we have

$$P_{\text{succ}} \geq \frac{1}{16}\left(\frac{1}{4} - M\epsilon_P\right)^2 \geq \frac{1}{16}\left(\frac{1}{4} - \frac{1}{162M}\right)^2 \geq \frac{1}{400},$$

since $M \geq 4 > 1$. We remark that because we mostly only care about the asymptotic complexity we did not bound this probability very tightly, and this bound may be a very loose one. The actual success probability can be much larger than this and can be further increased by optimizing the choice of M and ϵ_P.

Bibliography

[1] Nilin Abrahamsen. “A polynomial-time algorithm for ground states of spin trees”. In: *arXiv preprint arXiv:1907.04862* (2019).

[2] Nilin Abrahamsen et al. “Entanglement area law for 1D gauge theories and bosonic systems”. In: *arXiv preprint arXiv:2203.16012* (2022).

[3] Daniel S Abrams and Seth Lloyd. “Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors”. In: *Phys. Rev. Lett.* 83.24 (1999), p. 5162. DOI: `10.1103/PhysRevLett.83.5162`.

[4] Daniel S Abrams and Seth Lloyd. “Simulation of many-body Fermi systems on a universal quantum computer”. In: *Physical Review Letters* 79.13 (1997), p. 2586.

[5] Dorit Aharonov and Amnon Ta-Shma. “Adiabatic quantum state generation and statistical zero knowledge”. In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM. 2003, pp. 20–29.

[6] Dorit Aharonov et al. “The power of quantum systems on a line”. In: *Comm. Math. Phys.* 287.1 (2009), pp. 41–65.

[7] Yakir Aharonov and David Bohm. “Time in the quantum theory and the uncertainty relation for time and energy”. In: *Phys. Rev.* 122.5 (1961), p. 1649.

[8] Yakir Aharonov, Serge Massar, and Sandu Popescu. “Measuring energy, estimating Hamiltonians, and the time-energy uncertainty relation”. In: *Phys. Rev. A* 66.5 (2002), p. 052107. DOI: `10.1103/PhysRevA.66.052107`.

[9] Tameem Albash and Daniel A. Lidar. “Adiabatic quantum computation”. In: *Rev. Mod. Phys.* 90 (2018), p. 015002.

[10] Andris Ambainis. “On physical problems that are slightly more difficult than QMA”. In: *2014 IEEE 29th Conference on Computational Complexity (CCC)*. IEEE. 2014, pp. 32–43.

[11] Andris Ambainis. “Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations”. In: *arXiv preprint arXiv:1010.4458* (2010).

[12] Andris Ambainis. “Variable time amplitude amplification and quantum algorithms for linear algebra problems”. In: *STACS’12 (29th Symposium on Theoretical Aspects of Computer Science)*. Vol. 14. 2012, pp. 636–647.

[13] Dong An, Di Fang, and Lin Lin. “Time-dependent Hamiltonian Simulation of Highly Oscillatory Dynamics and Superconvergence for Schrödinger Equation”. In: *Quantum* 6 (2022), p. 690.

[14] Dong An, Di Fang, and Lin Lin. “Time-dependent unbounded Hamiltonian simulation with vector norm scaling”. In: *Quantum* 5 (2021), p. 459.

[15] Dong An and Lin Lin. “Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm”. In: *ACM Transactions on Quantum Computing* 3.2 (2022), pp. 1–28.

[16] Anurag Anshu, Itai Arad, and David Gosset. “An area law for 2D frustration-free spin systems”. In: *arXiv preprint arXiv:2103.02492* (2021).

[17] Anurag Anshu, Itai Arad, and David Gosset. “Entanglement subvolume law for 2D frustration-free spin systems”. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 868–874.

[18] Simon Apers, András Gilyén, and Stacey Jeffery. “A unified framework of quantum walk search”. In: *arXiv preprint arXiv:1912.04233* (2019).

[19] Simon Apers and Alain Sarlette. “Quantum fast-forwarding: Markov chains and graph property testing”. In: *Quantum Information & Computation* 19.3-4 (2019), pp. 181–213. URL: `https://dl.acm.org/doi/10.5555/3370245.3370246`.

[20] Itai Arad, Zeph Landau, and Umesh Vazirani. “Improved one-dimensional area law for frustration-free systems”. In: *Physical Review B* 85 (19 May 2012), p. 195145. DOI: `10.1103/PhysRevB.85.195145`.

[21] Itai Arad et al. “An area law and sub-exponential algorithm for 1D systems”. In: *arXiv preprint arXiv:1301.1162* (2013).

[22] Juan Miguel Arrazola et al. “Quantum-inspired algorithms in practice”. In: *arXiv preprint arXiv:1905.10415* (2019).

[23] Alán Aspuru-Guzik et al. “Simulated quantum computation of molecular energies”. In: *Science* 309.5741 (2005), pp. 1704–1707.

[24] Yosi Atia and Dorit Aharonov. "Fast-forwarding of Hamiltonians and exponentially precise measurements". In: *Nature Comm.* 8.1 (2017). DOI: `10.1038/s41467-017-01637-7`.

[25] Ryan Babbush et al. "Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation". In: *Physical Review A* 91.2 (2015), p. 022311.

[26] Ryan Babbush et al. "Encoding electronic spectra in quantum circuits with linear T complexity". In: *Physical Review X* 8.4 (2018), p. 041015.

[27] Ryan Babbush et al. "Focus beyond quadratic speedups for error-corrected quantum advantage". In: *PRX Quantum* 2.1 (2021), p. 010103.

[28] Ryan Babbush et al. "Low-depth quantum simulation of materials". In: *Phys. Rev. X* 8.1 (2018), p. 011044.

[29] AP Balachandran and SM Roy. "Quantum anti-Zeno paradox". In: *Physical review letters* 84.18 (2000), p. 4019. DOI: `10.1103/PhysRevLett.84.4019`.

[30] Johannes Bausch et al. "Undecidability of the spectral gap in one dimension". In: *arXiv preprint arXiv:1810.01858* (2018).

[31] Charles H Bennett et al. "Strengths and weaknesses of quantum computing". In: *SIAM J. Comput.* 26.5 (1997), pp. 1510–1523.

[32] D. W. Berry et al. "How to perform the most accurate possible phase measurements". In: *Phys. Rev. A* 80.5 (2009). DOI: `10.1103/physreva.80.052114`.

[33] Dominic W Berry and Andrew M Childs. "Black-box Hamiltonian simulation and unitary implementation". In: *arXiv preprint arXiv:0910.4157* (2009).

[34] Dominic W Berry, Andrew M Childs, and Robin Kothari. "Hamiltonian simulation with nearly optimal dependence on all parameters". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 792–809.

[35] Dominic W Berry et al. "Simulating Hamiltonian dynamics with a truncated Taylor series". In: *Phys. Rev. Lett.* 114.9 (2015), p. 090502.

[36] Dominic W. Berry et al. "Time-dependent Hamiltonian simulation with L1-norm scaling". In: *Quantum* 4 (2020), p. 254. DOI: `10.22331/q-2020-04-20-254`.

[37] Arvid J Bessen. "Lower bound for quantum phase estimation". In: *Physical Review A* 71.4 (2005), p. 042313.

[38] CLennart Bittel and Martin Kliesch. "Training variational quantum algorithms is NP-hard – even for logarithmically many qubits and free fermionic systems". In: *arXiv preprint arXiv:2101.07267* (2021).

[39] Sergio Boixo, Emanuel Knill, and Rolando D Somma. "Eigenpath traversal by phase randomization." In: *Quantum Info. Comput.* 9 (2009), pp. 833–855.

[40] Sergio Boixo and Rolando D Somma. "Parameter estimation with mixed-state quantum computation". In: *Physical Review A* 77.5 (2008), p. 052320.

[41] Kyle EC Booth et al. "Quantum-accelerated constraint programming". In: *arXiv preprint arXiv:2103.04502* (2021).

[42] Gilles Brassard et al. "Quantum amplitude amplification and estimation". In: *Contemp. Math.* 305 (2002), pp. 53–74.

[43] Carlos Bravo-Prieto et al. "Variational Quantum Linear Solver: A Hybrid Algorithm for Linear Systems". In: *arXiv:1909.05820* (2019).

[44] Ireneusz W Bulik, Gustavo E Scuseria, and Jorge Dukelsky. "Density matrix embedding from broken symmetry lattice mean fields". In: *Phys. Rev. B* 89.3 (2014), p. 035140. ISSN: 1098-0121. DOI: `10.1103/physrevb.89.035140`.

[45] Daniel Burgarth et al. "Non-Abelian phases from quantum Zeno dynamics". In: *Physical Review A* 88.4 (2013), p. 042107. DOI: `10.1103/PhysRevA.88.042107`.

[46] Earl Campbell. "Random Compiler for Fast Hamiltonian Simulation". In: *Phys. Rev. Lett.* 123.7 (2019). DOI: `10.1103/physrevlett.123.070503`.

[47] Earl T Campbell. "Early fault-tolerant simulations of the Hubbard model". In: *Quantum Science and Technology* 7.1 (2021), p. 015007.

[48] Yudong Cao et al. "Quantum algorithm and circuit design solving the Poisson equation". In: *New J. Phys.* 15.1 (2013), p. 013021.

[49] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. "The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation". In: *arXiv preprint arXiv:1804.01973* (2018).

[50] Rui Chao et al. "Finding Angles for Quantum Signal Processing with Machine Precision". In: (2020).

[51] Rui Chao et al. "Finding Angles for Quantum Signal Processing with Machine Precision". In: *arXiv preprint arXiv:2003.02831* (2020).

[52] Chi-Fang Chen et al. "Quantum simulation via randomized product formulas: Low gate complexity with accuracy guarantees". In: *arXiv preprint arXiv:2008.11751* (2020).

[53] Nai-Hui Chia, Han-Hsuan Lin, and Chunhao Wang. "Quantum-inspired sublinear classical algorithms for solving low-rank linear systems". In: *arXiv preprint arXiv:1811.04852* (2018).

[54] Andrew M Childs, John Preskill, and Joseph Renes. “Quantum information and precision measurement”. In: *J. Mod. Optics* 47.2-3 (2000), pp. 155–176.

[55] Andrew M Childs and Yuan Su. “Nearly optimal lattice simulation by product formulas”. In: *Physical review letters* 123.5 (2019), p. 050503.

[56] Andrew M Childs et al. “A Theory of Trotter Error”. In: *arXiv preprint arXiv:1912.08854* (2019).

[57] Andrew M Childs et al. “Quantum search by measurement”. In: *Phys. Rev. A* 66.3 (2002), p. 032314.

[58] Andrew M Childs et al. “Toward the first quantum simulation with quantum speedup”. In: *Proc. Natl. Acad. Sci* 115.38 (2018), pp. 9456–9461.

[59] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. “Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision”. In: *SIAM J. Comput.* 46 (2017), pp. 1920–1950.

[60] Andrew M. Childs et al. “Theory of Trotter Error with Commutator Scaling”. In: *Phys. Rev. X* 11.1 (2021). DOI: `10.1103/physrevx.11.011020`.

[61] Anirban Narayan Chowdhury, Yigit Subasi, and Rolando D Somma. “Improved implementation of reflection operators”. In: *arXiv preprint arXiv:1803.02466* (2018).

[62] Richard Cleve et al. “Quantum algorithms revisited”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 339–354.

[63] Pedro Costa et al. “Optimal scaling quantum linear systems solver via discrete adiabatic theorem”. In: *arXiv preprint arXiv:2111.08152* (2021).

[64] Marcus Cramer, Jens Eisert, and MB Plenio. “Statistics dependence of the entanglement entropy”. In: *Physical Review Letters* 98.22 (2007), p. 220603.

[65] Marcus Cramer et al. “Entanglement-area law for general bosonic harmonic lattice systems”. In: *Physical Review A* 73.1 (2006), p. 012309.

[66] Toby S Cubitt, David Perez-Garcia, and Michael M Wolf. “Undecidability of the spectral gap”. In: *Nature* 528.7581 (2015), pp. 207–211.

[67] Zhi-Hao Cui, Tianyu Zhu, and Garnet Kin-Lic Chan. “Efficient Implementation of Ab Initio Quantum Embedding in Periodic Systems: Density Matrix Embedding Theory”. In: *J. Chem. Theory Comput.* 16 (2019), pp. 119–129. ISSN: 1549-9618. DOI: `10.1021/acs.jctc.9b00933`.

[68] Zhi-Hao Cui et al. "Ground-state phase diagram of the three-band Hubbard model in various parametrizations from density matrix embedding theory". In: *arXiv preprint arXiv:2001.04951* (2020).

[69] Yulong Dong, Lin Lin, and Yu Tong. "Ground state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices". In: *arXiv preprint arXiv:2204.05955* (2022).

[70] Yulong Dong et al. "Efficient Phase Factor Evaluation in Quantum Signal Processing". In: *arXiv preprint arXiv:2002.11649* (2020).

[71] J. Eisert, M. Cramer, and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". In: *Reviews of Modern Physics* 82 (1 Feb. 2010), pp. 277–306. DOI: `10.1103/RevModPhys.82.277`.

[72] Alexander Elgart and George A Hagedorn. "A note on the switching adiabatic theorem". In: *J. Math. Phys.* 53.10 (2012), p. 102202.

[73] Joseph Emerson et al. "Estimation of the local density of states on a quantum computer". In: *Physical Review A* 69.5 (2004), p. 050305.

[74] Paul Erdös. "Some remarks on polynomials". In: *Bulletin of the American Mathematical Society* 53.12 (1947), pp. 1169–1176. DOI: `10.1090/S0002-9904-1947-08938-2`.

[75] Alexandre Eremenko and Peter Yuditskii. "Uniform approximation of $\operatorname{sgn}(x)$ by polynomials and entire functions". In: *Journal d'Analyse Mathématique* 101.1 (2007), pp. 313–324.

[76] P Facchi et al. "Berry phase from a quantum Zeno effect". In: *Physics Letters A* 257.5-6 (1999), pp. 232–240. DOI: `10.1016/S0375-9601(99)00323-0`.

[77] Paolo Facchi and Saverino Pascazio. "Quantum Zeno dynamics: mathematical and physical aspects". In: *Journal of Physics A: Mathematical and Theoretical* 41.49 (2008), p. 493001. DOI: `10.1088/1751-8113/41/49/493001`.

[78] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm". In: *arXiv preprint arXiv:1411.4028* (2014).

[79] Edward Farhi et al. "Quantum computation by adiabatic evolution". In: *arXiv: quant-ph/0001106* (2000).

[80] Yimin Ge, Jordi Tura, and J Ignacio Cirac. "Faster ground state preparation and high-precision ground energy estimation with fewer qubits". In: *J. Math. Phys.* 60.2 (2019), p. 022202.

[81] Yimin Ge, Jordi Tura, and J Ignacio Cirac. "Faster ground state preparation and high-precision ground energy estimation with fewer qubits". In: *J. Math. Phys.* 60.2 (2019), p. 022202. DOI: `10.1063/1.5027484`.

[82] Michael B Giles. "Multilevel Monte Carlo methods." In: *Acta Numer.* 24 (2015), pp. 259–328.

[83] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. "Optimizing quantum optimization algorithms via faster quantum gradient computation". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2019, pp. 1425–1444.

[84] András Gilyén, Seth Lloyd, and Ewin Tang. "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension". In: *arXiv preprint arXiv:1811.04909* (2018).

[85] András Gilyén et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: *arXiv preprint arXiv:1806.01838* (2018).

[86] András Gilyén et al. "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*. 2019, pp. 193–204. DOI: `10.1145/3313276.3316366`.

[87] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Advances in quantum metrology". In: *Nature Photon.* 5.4 (2011), p. 222. DOI: `10.1038/nphoton.2011.35`.

[88] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum Metrology". In: *Phys. Rev. Lett.* 96.1 (2006). DOI: `10.1103/physrevlett.96.010401`.

[89] Robert B Griffiths and Chi-Sheng Niu. "Semiclassical Fourier transform for quantum computation". In: *Phys. Rev. Lett.* 76.17 (1996), p. 3228. DOI: `10.1103/physrevlett.76.3228`.

[90] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[91] Lov K Grover. "Fixed-point quantum search". In: *Physical Review Letters* 95.15 (2005), p. 150501. DOI: `10.1103/PhysRevLett.95.150501`.

[92] Jeongwan Haah. "Product decomposition of periodic functions in quantum signal processing". In: *Quantum* 3 (2019), p. 190.

[93] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. In: *Phys. Rev. Lett.* 103 (2009), p. 150502.

[94] Matthew B Hastings. “An area law for one-dimensional quantum systems”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.08 (2007), P08024. DOI: `10.1088/1742-5468/2007/08/p08024`.

[95] B. L. Higgins et al. “Entanglement-free Heisenberg-limited phase estimation”. In: *Nature* 450.7168 (2007), pp. 393–396. DOI: `10.1038/nature06257`.

[96] William J Huggins et al. “A non-orthogonal variational quantum eigensolver”. In: *New J. of Phys.* 22.7 (2020), p. 073009. DOI: `10.1088/1367-2630/ab867b`.

[97] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. “Bounds for the adiabatic approximation with applications to quantum computation”. In: *J. Math. Phys.* 48.10 (2007), p. 102111.

[98] Zhang Jiang et al. “Quantum algorithms to simulate many-body physics of correlated fermions”. In: *Phys. Rev. Applied* 9.4 (2018), p. 044036.

[99] Stephen P Jordan, Keith SM Lee, and John Preskill. “Quantum algorithms for quantum field theories”. In: *Science* 336.6085 (2012), pp. 1130–1133.

[100] Julia Kempe, Alexei Kitaev, and Oded Regev. “The complexity of the local Hamiltonian problem”. In: *SIAM J. Comput.* 35.5 (2006), pp. 1070–1097.

[101] A Yu Kitaev. “Quantum measurements and the Abelian stabilizer problem”. In: *arXiv preprint quant-ph/9511026* (1995).

[102] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Graduate Studies in Mathematics 47. American Mathematical Soc., 2002. DOI: `10.1090/gsm/047`.

[103] Ian D Kivlichan et al. “Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via Trotterization”. In: *Quantum* 4 (2020), p. 296. DOI: `10.22331/q-2020-07-16-296`.

[104] Ian D Kivlichan et al. “Quantum simulation of electronic structure with linear depth and connectivity”. In: *Phys. Rev. Lett.* 120.11 (2018), p. 110501.

[105] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma. “Optimal quantum measurements of expectation values of observables”. In: *Phys. Rev. A* 75.1 (2007). DOI: `10.1103/PhysRevA.75.012328`.

[106] Gerald Knizia and Garnet Kin-Lic Chan. “Density matrix embedding: A simple alternative to dynamical mean-field theory”. In: *Phys. Rev. Lett.* 109.18 (2012), p. 186404. DOI: `10.1103/physrevlett.109.186404`.

[107] Gerald Knizia and Garnet Kin-Lic Chan. "Density matrix embedding: A strong-coupling quantum embedding theory". In: *J. Chem. Theory Comput.* 9.3 (2013), pp. 1428–1432. DOI: `10.1021/ct301044e`.

[108] David Layden. "First-Order Trotter Error from a Second-Order Perspective". In: *arXiv preprint arXiv:2107.08032* (2021).

[109] Jessica Lemieux et al. "Resource estimate for quantum many-body ground state preparation on a quantum computer". In: *arXiv preprint arXiv:2006.04650* (2020).

[110] Lin Lin and Yu Tong. "Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers". In: *PRX Quantum* 3.1 (2022), p. 010318.

[111] Lin Lin and Yu Tong. "Near-optimal ground state preparation". In: *Quantum* 4 (2020), p. 372. DOI: `10.22331/q-2020-12-14-372`.

[112] Lin Lin and Yu Tong. "Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems". In: *Quantum* 4 (2020), p. 361. DOI: `10.22331/q-2020-11-11-361`.

[113] Seth Lloyd. "Universal quantum simulators". In: *Science* (1996), pp. 1073–1078.

[114] Guang Hao Low and Isaac L Chuang. "Hamiltonian simulation by qubitization". In: *Quantum* 3 (2019), p. 163.

[115] Guang Hao Low and Isaac L. Chuang. "Optimal Hamiltonian Simulation by Quantum Signal Processing". In: *Phys. Rev. Lett.* 118 (2017), p. 010501.

[116] Guang Hao Low and Nathan Wiebe. "Hamiltonian simulation in the interaction picture". In: *arXiv preprint arXiv:1805.00675* (2018).

[117] Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. "Methodology of resonant equiangular composite quantum gates". In: *Phys. Rev. X* 6 (2016), p. 041067.

[118] Sirui Lu, Mari Carmen Bañuls, and J Ignacio Cirac. "Algorithms for quantum simulation at finite energies". In: *arXiv preprint arXiv:2006.03032* (2020).

[119] Sam McArdle et al. "Quantum computational chemistry". In: *Reviews of Modern Physics* 92.1 (2020), p. 015003.

[120] Jarrod R McClean et al. "The theory of variational hybrid quantum-classical algorithms". In: *New J. Phys.* 18.2 (2016), p. 023023. DOI: `10.1088/1367-2630/18/2/023023`.

[121] Baidyanath Misra and EC George Sudarshan. “The Zeno’s paradox in quantum theory”. In: *Journal of Mathematical Physics* 18.4 (1977), pp. 756–763. DOI: 10.1063/1.523304.

[122] Mario Motta et al. “Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution”. In: *Nature Phys.* 16.2 (2019), pp. 205–210. DOI: 10.1038/s41567-019-0704-4.

[123] Mario Motta et al. “Quantum imaginary time evolution, quantum lanczos, and quantum thermal averaging”. In: *arXiv preprint arXiv:1901.07653* (2019).

[124] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. “Fast amplification of QMA”. In: *Quantum Inf. Comput.* 9.11 (2009), pp. 1053–1068.

[125] Ashwin Nayak and Felix Wu. “The quantum query complexity of approximating the median and related statistics”. In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing.* 1999, pp. 384–393.

[126] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information.* 2002.

[127] Thomas E O’Brien et al. “Error mitigation via verified phase estimation”. In: *arXiv preprint arXiv:2010.02538* (2020).

[128] Thomas E O’Brien, Brian Tarasinski, and Barbara M Terhal. “Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments”. In: *New J. Phys.* 21.2 (2019), p. 023022.

[129] Peter JJ O’Malley et al. “Scalable quantum simulation of molecular energies”. In: *Phys. Rev. X* 6.3 (2016), p. 031007. DOI: 10.1103/PhysRevX.6.031007.

[130] Roberto Oliveira and Barbara M Terhal. “The complexity of quantum spin systems on a two-dimensional square lattice”. In: *arXiv preprint quant-ph/0504050* (2005).

[131] S. Östlund and S. Rommer. “Thermodynamic limit of density matrix renormalization”. In: *Phys. Rev. Lett.* 75.19 (1995), p. 3537.

[132] Robert M Parrish and Peter L McMahon. “Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation”. In: *arXiv preprint arXiv:1909.08925* (2019).

[133] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Comm.* 5.1 (2014). DOI: 10.1038/ncomms5213.

[134] David Poulin and Pawel Wocjan. “Preparing ground states of quantum many-body systems on a quantum computer”. In: *Phys. Rev. Lett.* 102.13 (2009), p. 130503. DOI: 10.1103/PhysRevLett.102.130503.

[135] David Poulin and Pawel Wocjan. “Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer”. In: *Phys. Rev. Lett.* 103.22 (2009). DOI: 10.1103/physrevlett.103.220502.

[136] David Poulin et al. “Quantum algorithm for spectral measurement with a lower gate count”. In: *Physical review letters* 121.1 (2018), p. 010501. DOI: 10.1103/PhysRevLett.121.010501.

[137] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.

[138] Eugene Remes. “Sur le calcul effectif des polynomes d’approximation de Tchebichef”. In: *CR Acad. Sci. Paris* 199 (1934), pp. 337–340.

[139] Eugene Y Remez. “Sur la détermination des polynômes d’approximation de degré donnée”. In: *Comm. Soc. Math. Kharkov* 10.196 (1934), pp. 41–63.

[140] Chang-Han Rhee and Peter W Glynn. “A new approach to unbiased estimation for SDE’s”. In: *Proceedings of the 2012 Winter Simulation Conference (WSC).* IEEE. 2012, pp. 1–7. DOI: 10.1109/WSC.2012.6465150.

[141] Chang-Han Rhee and Peter W Glynn. “Unbiased estimation with square root convergence for SDE models”. In: *Oper. Res.* 63.5 (2015), pp. 1026–1043. DOI: 10.1287/opre.2015.1404.

[142] AE Russo et al. “Evaluating energy differences on a quantum computer with robust phase estimation”. In: *arXiv preprint arXiv:2007.08697* (2020).

[143] Yousef Saad. *Iterative methods for sparse linear systems.* Vol. 82. SIAM, 2003.

[144] Sushant Sachdeva and Nisheeth K Vishnoi. “Faster algorithms via approximation theory”. In: *Theoretical Computer Science* 9.2 (2013), pp. 125–210. DOI: 10.1561/0400000065.

[145] Yuval R Sanders et al. “Compilation of fault-tolerant quantum heuristics for combinatorial optimization”. In: *PRX Quantum* 1.2 (2020), p. 020312.

[146] U. Schollwöck. “The density-matrix renormalization group in the age of matrix product states”. In: *Ann. Phys.* 326.1 (2011), pp. 96–192.

[147] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332.

[148] Rolando Somma et al. “Simulating physical phenomena by quantum networks”. In: *Phys. Rev. A* 65.4 (2002), p. 042323.

[149] Rolando D Somma. “Quantum eigenvalue estimation via time series analysis”. In: *New J. Phys.* 21.12 (2019), p. 123025. DOI: 10.1088/1367-2630/ab5c60.

[150] Rolando D Somma et al. "Quantum simulations of classical annealing processes". In: *Physical review letters* 101.13 (2008), p. 130504. DOI: 10.1103/PhysRevLett.101.130504.

[151] Nicholas H. Stair, Renke Huang, and Francesco A. Evangelista. "A Multireference Quantum Krylov Algorithm for Strongly Correlated Electrons". In: *J. Chem. Theory Comp.* 16.4 (Feb. 2020), pp. 2236–2245. DOI: 10.1021/acs.jctc.9b01125.

[152] Yuan Su, Hsin-Yuan Huang, and Earl T Campbell. "Nearly tight Trotterization of interacting electrons". In: *arXiv preprint arXiv:2012.09194* (2020).

[153] Yuan Su et al. "Fault-tolerant quantum simulations of chemistry in first quantization". In: *PRX Quantum* 2.4 (2021), p. 040332.

[154] Yiğit Subaşı, Rolando D. Somma, and Davide Orsucci. "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing". In: *Phys. Rev. Lett.* 122 (2019), p. 060504.

[155] Kenji Sugisaki et al. "Quantum chemistry on quantum computers: A method for preparation of multiconfigurational wave functions on quantum computers without performing post-hartree–fock calculations". In: *ACS central science* 5.1 (2018), pp. 167–175.

[156] Chong Sun et al. "Finite-temperature density matrix embedding theory". In: *arXiv preprint arXiv:1911.07439* 101 (2019). ISSN: 2469-9950. DOI: 10.1103/physrevb.101.075131.

[157] Masuo Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics". In: *J. Math. Phys.* 32.2 (1991), pp. 400–407. DOI: 10.1063/1.529425.

[158] Mario Szegedy. "Quantum speed-up of Markov chain based algorithms". In: *45th Annual IEEE symposium on foundations of computer science*. IEEE. 2004, pp. 32–41. DOI: 10.1109/FOCS.2004.53.

[159] Ewin Tang. "A quantum-inspired classical algorithm for recommendation systems". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 217–228.

[160] Ewin Tang. "Quantum-inspired classical algorithms for principal component analysis and supervised clustering". In: *arXiv preprint arXiv:1811.00414* (2018).

[161] Yu Tong et al. "Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions". In: *Physical Review A* 104.3 (2021), p. 032422.

[162] Yu Tong et al. "Provably accurate simulation of gauge theories and bosonic systems". In: *arXiv preprint arXiv:2110.06942* (2021).

[163] Minh C Tran et al. "Destructive error interference in product-formula lattice simulation". In: *Physical review letters* 124.22 (2020), p. 220502.

[164] Takashi Tsuchimochi, Matthew Welborn, and Troy Van Voorhis. "Density matrix embedding in an antisymmetrized geminal power bath". In: *J. Chem. Phys* 143.2 (2015), p. 024107. DOI: `10.1063/1.4926650`.

[165] Norm M Tubman et al. "Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices". In: *arXiv preprint arXiv:1809.05523* (2018).

[166] Kianna Wan, Mario Berta, and Earl T Campbell. "A randomized quantum algorithm for statistical phase estimation". In: *arXiv preprint arXiv:2110.12071* (2021).

[167] Daochen Wang, Oscar Higgott, and Stephen Brierley. "Accelerated variational quantum eigensolver". In: *Phys. Rev. Lett.* 122.14 (2019), p. 140504. DOI: `10.1103/physrevlett.122.140504`.

[168] Jiasu Wang, Yulong Dong, and Lin Lin. "On the energy landscape of symmetric quantum signal processing". In: *arXiv preprint arXiv:2110.04993* (2021).

[169] S. R. White. "Density matrix formulation for quantum renormalization groups". In: *Phys. Rev. Lett.* 69.19 (1992), p. 2863.

[170] Nathan Wiebe et al. "Bayesian inference via rejection filtering". In: *arXiv preprint arXiv:1511.06458* (2015).

[171] Pawel Wocjan and Anura Abeyesinghe. "Speedup via quantum sampling". In: *Physical Review A* 78.4 (2008), p. 042336. DOI: `10.1103/PhysRevA.78.042336`.

[172] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. "Quantum Linear System Algorithm for Dense Matrices". In: *Phys. Rev. Lett.* 120.5 (2018), p. 050502.

[173] Sebastian Wouters et al. "A practical guide to density matrix embedding theory in quantum chemistry". In: *J. Chem. Theory Comput.* 12.6 (2016), pp. 2706–2719. DOI: `10.1021/acs.jctc.6b00316`.

[174] Xiaojie Wu et al. "Enhancing robustness and efficiency of density matrix embedding theory via semidefinite programming and local correlation potential fitting". In: *Physical Review B* 102.8 (2020), p. 085123.

[175] Xiaojie Wu et al. "Projected Density Matrix Embedding Theory with Applications to the Two-Dimensional Hubbard Model". In: *J. Chem. Phys* 151 (2019), p. 064108. ISSN: 0021-9606. DOI: `10.1063/1.5108818`.

[176] Xiaosi Xu et al. "Variational algorithms for linear algebra". In: *arXiv:1909.03898* (2019).

[177] Changhao Yi and Elizabeth Crosson. "Spectral Analysis of Product Formulas for Quantum Simulation". In: *arXiv preprint arXiv:2102.12655* (2021).

[178] Lexing Ying. "Stable factorization for phase factors of quantum signal processing". In: *arXiv preprint arXiv:2202.02671* (2022).

[179] Theodore J Yoder, Guang Hao Low, and Isaac L Chuang. "Fixed-point quantum search with an optimal number of queries". In: *Physical review letters* 113.21 (2014), p. 210501. DOI: `10.1103/PhysRevLett.113.210501`.

[180] Ruizhe Zhang, Guoming Wang, and Peter Johnson. "Computing ground state properties with early fault-tolerant quantum computers". In: *arXiv preprint arXiv:2109.13957* (2021).

[181] Marcin Zwierz, Carlos A Pérez-Delgado, and Pieter Kok. "Ultimate limits to quantum metrology and the meaning of the Heisenberg limit". In: *Phys. Rev. A* 85.4 (2012), p. 042112. DOI: `10.1103/PhysRevA.85.042112`.

[182] Marcin Zwierz, Carlos A. Pérez-Delgado, and Pieter Kok. "General Optimality of the Heisenberg Limit for Quantum Metrology". In: *Phys. Rev. Lett.* 105.18 (2010). DOI: `10.1103/physrevlett.105.180402`.