Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems

Lin Lin[1,2] and Yu Tong[1]

[1]Department of Mathematics, University of California, Berkeley, CA 94720, USA

[2]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

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.

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. All QLSP solvers share the desirable property that the complexity with respect to the matrix dimension can be as low as $\mathcal{O}(\mathrm{polylog}(N))$, 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 [5, 14, 16, 17, 21, 34, 38, 56, 61, 62]. 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 $\kappa = \|A\|\|A^{-1}\|$) [5, 56], 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 [33], 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) [34, 43], 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 BQP=PSPACE [38]. 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 8) combines quantum eigenstate filtering with the time-optimal adiabatic quantum computing (AQC) approach [5]. 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 11) combines quantum eigenstate filtering with the time-optimal version of the approach based on the quantum Zeno effect (QZE) [12, 56]. 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 [45] 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 [40], which relies on Hamiltonian simulation [10, 11, 42, 43, 44, 45] and the quantum Fourier transform. We treat the Hamiltonian simulation $e^{-iH\tau}$ with some fixed τ as an oracle called U_{sim},

where τ satisfies $\tau\|H\| < \pi$. Ref. [30, Appendix B] contains a very detailed analysis of the complexities of using phase estimation together with amplitude amplification. From the analysis in Ref. [30], 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 [30, 48, 49]. The filtering method developed by Poulin and Wocjan [48] (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 $al.$ [30, 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. [49].

Ge et $al.$ [30] also proposed two eigenstate filtering algorithms using linear combination of unitaries (LCU) [11, 21], 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 2. There are several other hybrid quantum-classical algorithms to compute ground state energy and to prepare the ground state [47, 55], 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 [38] 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 [52].

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) [11, 21] and quantum signal processing (QSP) or quantum singular value transformation (QSVT) [34, 43] techniques can reduce the query complexity to $\mathcal{O}(\kappa^2\,\text{polylog}(\kappa/\epsilon))$. 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) [4] technique, and the resulting query complexity for solving QLSP is $\mathcal{O}(\kappa\,\text{polylog}(\kappa/\epsilon)))$ [17, 21]. 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.

The recently developed randomization method (RM) [56] is the first algorithm that yields near-optimal scaling with respect to κ, without using techniques such as VTAA.

Algorithm	Query complexity	Remark
HHL [38]	$\widetilde{\mathcal{O}}(\kappa^2/\epsilon)$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa/\epsilon^3)$ [3]
Linear combination of unitaries (LCU) [17, 21]	$\widetilde{\mathcal{O}}(\kappa^2 \operatorname{polylog}(1/\epsilon))$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$
Quantum singular value transformation (QSVT) [34]	$\widetilde{\mathcal{O}}(\kappa^2 \log(1/\epsilon))$	
Randomization method (RM) [56]	$\widetilde{\mathcal{O}}(\kappa/\epsilon)$	w. repeated phase estimation, complexity becomes $\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$
Time-optimal adiabatic quantum computing (AQC(exp)) [5]	$\widetilde{\mathcal{O}}(\kappa \operatorname{polylog}(1/\epsilon))$	No need for any amplitude amplification. Use time-dependent Hamiltonian simulation.
Eigenstate filtering+AQC (Theorem 8)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification.
Eigenstate filtering+QZE (Theorem 11)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification. Does not rely on any complex subroutines.

Table 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 [10, 11, 43, 44, 45]. The LCU method [21] and the gate-based implementation of the RM method [56] 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. [17]. Time complexities and gate complexities are converted to query complexities with respect to the oracles in this paper. [32, 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.

RM was inspired by adiabatic quantum computation (AQC) [2, 28, 39], 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 [5] 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) [29] can be as low as $\mathcal{O}(\kappa \operatorname{polylog}(1/\epsilon))$ [5]. 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 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 [58, 59] 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 [19, 31], 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 [8]. 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 $f = \mathcal{O}(g\,\mathrm{polylog}(g))$.

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 2 we briefly review block-encoding and QSP, as well as using QSP to directly solve QLSP with a non-optimal complexity. In Section 3 we introduce the minimax polynomial we are using and our eigenstate filtering method based on it. In Section 4 we combine eigenstate filtering with AQC to solve the QLSP. In Section 5 we present another method to solve the QLSP using QZE and eigenstate filtering. In Section 6 we discuss some practical aspects of our algorithms and future work.

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

$$\|A - \alpha(\langle 0^m| \otimes I)U(|0^m\rangle \otimes I)\| \leq \epsilon. \tag{1}$$

Another way to express Eq. (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 [21, Section 4.1] and [11, Lemma 10] for details, as well as [32, 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 1. (Polynomial eigenvalue transformation via quantum signal processing[1] [34, Theorem 31]): *Let U be an (α, m, ϵ)-block-encoding of a Hermitian 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 1 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 [34, 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 [21]. 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 [32, 37]. There are however difficulties in computing such phase factors, which will be discussed in Section 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 D). 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

---

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

$\mathcal{D}_{1/(\alpha\kappa)}$ for $c = 4\alpha\kappa/3$. Note that ϵ' is the accuracy of the polynomial approximation, so that the unnormalized state $P(A/\alpha)\,|b\rangle$ would differ from the desired $(\alpha/c)A^{-1}\,|b\rangle$ by ϵ'. In order to obtain a normalized solution $P(A/\alpha)\,|b\rangle\,/\|P(A/\alpha)\,|b\rangle\|$ that is ϵ-close to the normalized solution $|x\rangle = A^{-1}\,|b\rangle\,/\|A^{-1}\,|b\rangle\|$, we first note that $\|A^{-1}\,|b\rangle\| \geq 1$. So the normalization would amplify the error by a factor of approximately $c/(\alpha\|A^{-1}\,|b\rangle\|) \leq 4\kappa/3$. 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 [32, Corollary 69]. Then by Theorem 1' we have a circuit $\widetilde{U}$ satisfying

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

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

As $\|A^{-1}\,|b\rangle\| \geq 1$, the probability of success is $\Omega(1/\kappa^2)$. Using amplitude amplification [13], 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 [21].

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 A. 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 [52, Theorem 6.25]. A plot of the polynomial is given in Fig. 1. $R_\ell(x; \Delta)$ has several nice properties:

Lemma 2. *(i)* $R_\ell(x; \Delta)$ *solves the minimax problem*

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

(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* $R_\ell(0; \Delta) = 1$.
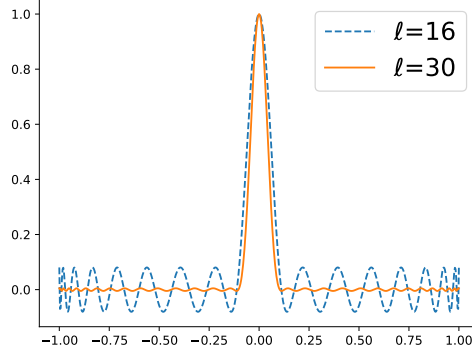*(iii)* $|R_\ell(x; \Delta)| \leq 1$ *for all* $|x| \leq 1$.

Figure 1: The polynomial $R_\ell(x, \Delta)$ for $\ell = 16$ and 30, $\Delta = 0.1$.

A proof of the above lemma is provided in Appendix E. If we apply this polynomial to $H - \lambda I$, Lemma 2 (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 A). The gap separating 0 from other eigenvalues of $\widetilde{H}$ is lower bounded by $\widetilde{\Delta} = \Delta/2\alpha$, as explained at the 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 2. 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 2, 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 3. (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 3 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 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 [21].

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 [13, 35]. This problem has been considered in Ref. [22].

For a given Hamiltonian H, with the same assumptions as in Theorem 3, 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 B, we can implement the reflection operator R_λ through QSP. The cost is summarized as follows:

Theorem 4. *Under the same assumption as Theorem 3, 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 B. This reflection operator further enables us to construct a block-encoding of the θ-reflection operator.

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

This operator is useful in fixed-point amplitude amplification [36, 63]. The cost is summarized as follows:

Corollary 5. *Under the same assumption as Theorem 3, a $(1, m + 3, \epsilon)$-block-encoding of $P_\lambda + e^{i\theta}(I - P_\lambda)$, 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 B.

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 [34, Lemma 14], together with constant shift. The filtering polynomial for this kind of task can also be obtained numerically through Remez algorithm [51], followed by a optimization based procedure to efficiently identify the phase factors. For more details we refer readers to Ref. [23].

Remark 6. *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 [52, Eq. (6.113)]. This is useful for obtaining the stationary distribution of an ergodic and reversible Markov chain because the discriminant matrix [6, 7, 57] 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 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

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

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

$$O_B |0\rangle = |b\rangle. \tag{3}$$

This is the same as the assumption used in [21, 56]. The oracles can be used to construct a $(d, n + 2, 0)$-block-encoding of A [11, 21].

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. [21] it is assumed that that $\|A\| = 1$ and the condition number is exactly κ [21, Problem 1], which is slightly stronger than the assumption we are currently using.

Remark 7. *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 D, 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 D, which uses different Hamiltonians but only requires minor modifications. We define

$$H_1 = \begin{pmatrix} 0 & AQ_b \\ Q_b A & 0 \end{pmatrix} = |0\rangle \langle 1| \otimes AQ_b + |1\rangle \langle 0| \otimes Q_b A, \tag{4}$$

where $Q_b = I - |b\rangle \langle b|$. This Hamiltonian has been used in Refs. [5, 56]. As discussed in Appendix A, 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 $1/\kappa$ [5, 56]. 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.

4.1 Choosing the eigenpath

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

$$H_0 = \begin{pmatrix} 0 & Q_b \\ Q_b & 0 \end{pmatrix} = \sigma_x \otimes Q_b. \tag{5}$$

and

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

where H_1 is defined in Eq. (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) and (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

$$\Delta_*(f) = 1 - f + \frac{f}{\kappa}. \tag{6}$$

A proof can be found in [5].

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

$$((1 - f)I + fA) |x(f)\rangle \propto |b\rangle . \tag{7}$$

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.

$$\langle x(f)|\partial_f |x(f)\rangle = 0, \tag{8}$$

---

[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.

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 F. Furthermore, we have the estimate

$$\| \partial_f |x(f)\rangle \| \leq \frac{2}{\Delta_*(f)}. \tag{9}$$

The derivation of the existence and uniqueness of the differentiable eigenpath, together with the estimate (9) are given in Appendix F.

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 (9) we have

$$L \leq \int_0^1 \frac{2}{\Delta_*(f)} \mathrm{d}f = \frac{2\log(\kappa)}{1 - 1/\kappa}. \tag{10}$$

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

$$L(a, b) = \int_a^b \| \partial_f |x(f)\rangle \| \mathrm{d}f \leq \frac{2}{1 - 1/\kappa} \log\left(\frac{1 - (1 - 1/\kappa)a}{1 - (1 - 1/\kappa)b} \right) =: L_*(a, b). \tag{11}$$

4.2 Time-optimal adiabatic quantum computing

Here we briefly review the procedure of solving QLSP using the recently developed time-optimal AQC [5] and the eigenpath described in the previous section that has been used in [5, 56].

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 $1/\kappa$ [5, 56]. The time complexity of vanilla AQC is at least $T \sim \mathcal{O}(\kappa^2/\epsilon)$ [2, 5, 24, 39].

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 [5]. The first method is called AQC(p). For $1 < p < 2$, AQC(p) adopts the schedule

$$f(s) = \frac{\kappa}{\kappa - 1} \left[1 - \left(1 + s(\kappa^{p-1} - 1) \right)^{\frac{1}{1-p}} \right].$$ (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 [45]. 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 C, and the query complexity for implementing AQC(p) on a gate-based quantum computer is $\widetilde{\mathcal{O}}(\kappa/\epsilon)$.

4.3 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), through the block-encoding U_{H_1} constructed in Appendix A we only need an initial state for eigenstate filtering (note that this is not the initial state of the AQC time-evolution):

$$|\widetilde{x}_0\rangle = \gamma_0 |0\rangle |x\rangle + \gamma_1 |1\rangle |b\rangle + |\perp\rangle$$ (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 [45], 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. (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 C.

Then we can run the eigenstate filtering algorithm described in Section 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 3. 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 C). 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 D). As discussed in Remark 7, 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 8. *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 (2), and $|b\rangle$ is given by an oracle O_B in (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 $\text{poly}(n)$ 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 9. *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 11 which we are going to introduce later.*

We present numerical results obtained on a classical computer in Fig. 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. 2 shows the relation between η, κ, and ℓ obtained in the numerical experiment.
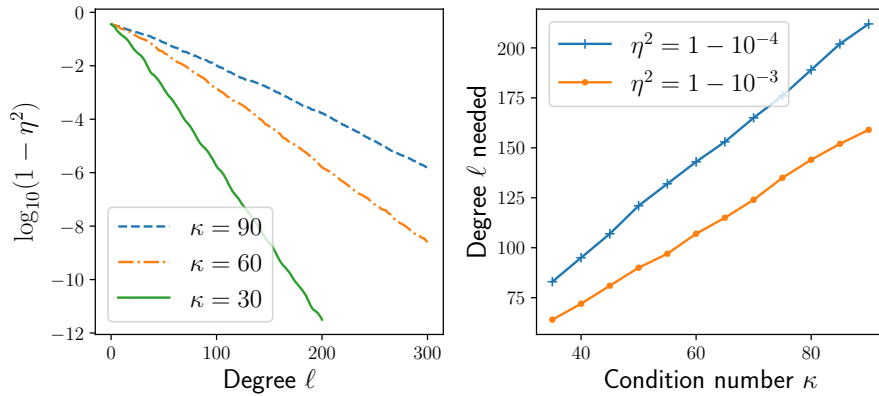
Figure 2: Left: fidelity η converges to 1 exponentially as ℓ in the eigenvalues filtering algorithm increases, for different κ. Right: the smallest ℓ needed to achieve fixed fidelity η grows linearly with respect to condition number κ. The initial state in eigenstate filtering is prepared by running AQC(p) for $T = 0.2\kappa$, with $p = 1.5$, which achieves an initial fidelity of about 0.6.

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 [9, 15, 26, 27, 46]. A variant of QZE [12, Lemma 1] can be viewed as a particular way for implementing adiabatic quantum computing [41, 50, 54], 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 [12, 56].

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. (7) and (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 [12, 20, 56]. 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. 4.1, and to ensure the undesired part of the null space does not interfere with our computation.

5.1 The algorithm

As in Section 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

$$f(s) = \frac{1 - \kappa^{-s}}{1 - \kappa^{-1}} \tag{14}$$

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 [20]. This scheduling is chosen so that

$$L(f_j, f_{j+1}) \leq L_*(f_j, f_{j+1}) = \frac{2 \log(\kappa)}{M(1 - 1/\kappa)}, \tag{15}$$

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 A, we can construct $(1 - f + fd, n + 6, 0)$-block-encoding for each $H(f)$, denoted by $U_H(f)$. This construction uses [34, 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 A) , 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 A. Applying the eigenstate filtering procedure in Section 3 to precision ϵ_P gives us an $(1, n + 7, \epsilon_P)$-block-encoding of

$$\bar{P}_0(f) = |0\rangle|x(f)\rangle\langle 0|\langle x(f)| + |1\rangle|b\rangle\langle 1|\langle b|, \tag{16}$$

which we denote by $U_{P_0}(f)$. By Theorem 3 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 3 is redundant because we do not need to shift by a multiple of the identity matrix. By definition of block-encoding we have

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

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

$$P_0(f) = |x(f)\rangle\langle x(f)|, \tag{17}$$

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

$$\widetilde{P}_0(f) = (\langle 0^{n+7}|\langle 0| \otimes I_n)U_{P_0}(f)(|0^{n+7}\rangle|0\rangle \otimes I_n) \tag{18}$$

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

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

$$\leq \left\| (\langle 0^{n+7}| \otimes I_1 \otimes I_n) U_{P_0}(f) (|0^{n+7}\rangle \otimes I_1 \otimes I_n) - \bar{P}_0(f) \right\|$$

$$\leq \epsilon_P.$$

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

As discussed in Section 4.1, 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. 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 2, 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:

$$|\widetilde{x}(f_j)\rangle = \frac{\widetilde{P}_0(f_j) |\widetilde{x}(f_{j-1})\rangle}{\|\widetilde{P}_0(f_j) |\widetilde{x}(f_{j-1})\rangle \|}. \tag{19}$$

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 G 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 10 (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 15, 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*

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

5.2 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 G. Under this assumption we have

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

Since

$$|\langle x(f_j)|x(f_{j-1})\rangle| \geq 1 - \frac{1}{2}\| |x(f_{j-1})\rangle - |x(f_j)\rangle \|^2, \tag{20}$$

$$\| |x(f_{j-1})\rangle - |x(f_j)\rangle \| \leq L(f_{j-1}, f_j) \leq L_*(f_{j-1}, f_j), \tag{21}$$

we have

$$p_{\text{success}} \geq \left(\prod_{j=1}^{M} \left(1 - \frac{1}{2}\| |x(f_{j-1})\rangle - |x(f_j)\rangle \|^2 \right) \right)^2$$

$$\geq \left(1 - \frac{2\log^2(\kappa)}{M^2(1 - 1/\kappa)^2} \right)^{2M}$$

$$\geq \left(1 - \frac{2\log^2(\kappa)}{M(1 - 1/\kappa)^2} \right)^2$$

$$\geq \frac{1}{4},$$

where the we have used Eq. (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 G.

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 G 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,

$$
\begin{aligned}
|\langle x|\widetilde{x}(1)\rangle| &= |\langle \widetilde{x}(f_M)|x(f_M)\rangle| \\
&= \frac{|\langle \widetilde{x}(f_{M-1})|\widetilde{P}_0(f_M)|x(f_M)\rangle|}{\|\widetilde{P}_0(f_M)|\widetilde{x}(f_{M-1})\rangle\|} \\
&\geq \frac{|\langle \widetilde{x}(f_{M-1})|P_0(f_M)|x(f_M)\rangle| - \epsilon/4}{\|P_0(f_M)|\widetilde{x}(f_{M-1})\rangle\| + \epsilon/4} \\
&= \frac{|\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle| - \epsilon/4}{|\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle| + \epsilon/4} \\
&\geq 1 - \frac{\epsilon/2}{|\langle \widetilde{x}(f_{M-1})|x(f_M)\rangle|} \\
&\geq 1 - \epsilon.
\end{aligned}
\tag{22}
$$

The derivation is similar to that of Eq. (A10), 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 G 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

$$
\begin{aligned}
\log\left(\frac{1}{\epsilon_P}\right) \sum_{j=1}^{M-1} \frac{1 - f(s_j) + f(s_j)d}{1 - f(s_j) + f(s_j)/\kappa} &\leq \log\left(\frac{1}{\epsilon_P}\right) M \int_0^1 \frac{1 - f(s) + f(s)d}{1 - f(s) + f(s)/\kappa} \mathrm{d}s \\
&= \log\left(\frac{1}{\epsilon_P}\right) M \left(\frac{d\kappa - 1}{\log(\kappa)} - \frac{d-1}{1 - 1/\kappa}\right),
\end{aligned}
\tag{23}
$$

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 5.1, 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 (23), and using the fact $\epsilon_P = \mathcal{O}(1/M^2)$, gives us the total query complexity of a single run

$$\mathcal{O}\left(d\kappa\left(\log(\kappa)\log\log(\kappa) + \log(1/\epsilon)\right)\right). \tag{24}$$

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. (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 D. The generalization to non-Hermitian matrices is the same as for Theorem 8, and it can be found in Appendix D as well. All other procedures are almost exactly the same. We summarize the results in the following theorem:

Theorem 11. *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 (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 7. Just like in the case of AQC-based QLSP algorithm, here if we have $\mathcal{O}(\text{poly}(n))$ gate complexity for the oracles $O_{A,1}$, $O_{A,2}$, and O_B, then the total gate complexity will be $\tilde{\mathcal{O}}(\text{poly}(n)d\kappa \log(1/\epsilon))$. 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 [60, 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.

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 $P|\widetilde{x}_0\rangle$, where P is an approximate projection operator, and $P|\widetilde{x}_0\rangle$ 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 $|\widetilde{x}_0\rangle$ 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 C). 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 [32, 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 [37], 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 [18, 23]. 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.

Acknowledgements

References

[1] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 2003. DOI: 10.1145/780542.780546.

[2] T. Albash and D. A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90: 015002, 2018. DOI: 10.1103/RevModPhys.90.015002.

[3] A. Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458*, 2010.

[4] A. Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 636–647, 2012.

[5] D. An and L. Lin. Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm. *arXiv:1909.05500*, 2019.

[6] S. Apers and A. Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information & Computation*, 19(3-4):181–213, 2019. URL https://dl.acm.org/doi/10.5555/3370245.3370246.

[7] S. Apers, A. Gilyén, and S. Jeffery. A unified framework of quantum walk search. *arXiv preprint arXiv:1912.04233*, 2019.

[8] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd. Quantum-inspired algorithms in practice. *arXiv preprint arXiv:1905.10415*, 2019.

[9] A. Balachandran and S. Roy. Quantum anti-Zeno paradox. *Physical review letters*, 84 (18):4019, 2000. DOI: 10.1103/PhysRevLett.84.4019.

[10] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114(9):090502, 2015. DOI: 10.1103/PhysRevLett.114.090502.

[11] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809. IEEE, 2015. DOI: 10.1109/FOCS.2015.54.

[12] S. Boixo, E. Knill, and R. D. Somma. Eigenpath traversal by phase randomization. *Quantum Info. Comput.*, 9:833–855, 2009. URL https://dl.acm.org/doi/10.5555/2011804.2011811.

[13] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Contemp. Math.*, 305:53–74, 2002. DOI: 10.1090/conm/305/05215.

[14] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles. Variational quantum linear solver: A hybrid algorithm for linear systems. *arXiv:1909.05820*, 2019.

[15] D. Burgarth, P. Facchi, V. Giovannetti, H. Nakazato, S. Pascazio, and K. Yuasa. Non-abelian phases from quantum Zeno dynamics. *Physical Review A*, 88(4):042107, 2013. DOI: 10.1103/PhysRevA.88.042107.

[16] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais. Quantum algorithm and circuit design solving the poisson equation. *New J. Phys.*, 15(1):013021, 2013. DOI: 10.1088/1367-2630/15/1/013021.

[17] S. Chakraborty, A. Gilyén, and S. Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. *arXiv preprint arXiv:1804.01973*, 2018.

[18] R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*, 2020.

[19] N.-H. Chia, H.-H. Lin, and C. Wang. Quantum-inspired sublinear classical algorithms for solving low-rank linear systems. *arXiv preprint arXiv:1811.04852*, 2018.

[20] A. M. Childs, E. Deotto, E. Farhi, J. Goldstone, S. Gutmann, and A. J. Landahl. Quantum search by measurement. *Phys. Rev. A*, 66(3):032314, 2002. DOI: 10.1103/PhysRevA.66.032314.

[21] A. M. Childs, R. Kothari, and R. D. Somma. Quantum algorithm for systems of linear

equations with exponentially improved dependence on precision. *SIAM J. Comput.*, 46:1920–1950, 2017. DOI: 10.1137/16M1087072.

[22] A. N. Chowdhury, Y. Subasi, and R. D. Somma. Improved implementation of reflection operators. *arXiv preprint arXiv:1803.02466*, 2018.

[23] Y. Dong, X. Meng, K. B. Whaley, and L. Lin. Efficient phase factor evaluation in quantum signal processing. *arXiv preprint arXiv:2002.11649*, 2020.

[24] A. Elgart and G. A. Hagedorn. A note on the switching adiabatic theorem. *J. Math. Phys.*, 53(10):102202, 2012. DOI: 10.1063/1.4748968.

[25] P. Erdös. Some remarks on polynomials. *Bulletin of the American Mathematical Society*, 53(12):1169–1176, 1947. DOI: 10.1090/S0002-9904-1947-08938-2.

[26] P. Facchi and S. Pascazio. Quantum Zeno dynamics: mathematical and physical aspects. *Journal of Physics A: Mathematical and Theoretical*, 41(49):493001, 2008. DOI: 10.1088/1751-8113/41/49/493001.

[27] P. Facchi, A. Klein, S. Pascazio, and L. Schulman. Berry phase from a quantum Zeno effect. *Physics Letters A*, 257(5-6):232–240, 1999. DOI: 10.1016/S0375-9601(99)00323-0.

[28] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[29] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[30] Y. Ge, J. Tura, and J. I. Cirac. Faster ground state preparation and high-precision ground energy estimation with fewer qubits. *J. Math. Phys.*, 60(2):022202, 2019. DOI: 10.1063/1.5027484.

[31] A. Gilyén, S. Lloyd, and E. Tang. Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension. *arXiv preprint arXiv:1811.04909*, 2018.

[32] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *arXiv preprint arXiv:1806.01838*, 2018. DOI: 10.1145/3313276.3316366.

[33] A. Gilyén, S. Arunachalam, and N. Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1425–1444, 2019. DOI: 10.1137/1.9781611975482.87.

[34] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. DOI: 10.1145/3313276.3316366.

[35] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996. DOI: 10.1145/237814.237866.

[36] L. K. Grover. Fixed-point quantum search. *Physical Review Letters*, 95(15):150501, 2005. DOI: 10.1103/PhysRevLett.95.150501.

[37] J. Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, 2019. DOI: 10.22331/q-2019-10-07-190.

[38] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009. DOI: 10.1007/978-3-642-27848-8_771-1.

[39] S. Jansen, M.-B. Ruskai, and R. Seiler. Bounds for the adiabatic approximation with

applications to quantum computation. *J. Math. Phys.*, 48(10):102111, 2007. DOI: 10.1063/1.2798382.

[40] A. Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[41] J. Lemieux, G. Duclos-Cianci, D. Sénéchal, and D. Poulin. Resource estimate for quantum many-body ground state preparation on a quantum computer. *arXiv preprint arXiv:2006.04650*, 2020.

[42] S. Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996. DOI: 10.1126/science.273.5278.1073.

[43] G. H. Low and I. L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, 2017. DOI: 10.1103/PhysRevLett.118.010501.

[44] G. H. Low and I. L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3: 163, 2019. DOI: 10.22331/q-2019-07-12-163.

[45] G. H. Low and N. Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint arXiv:1805.00675*, 2018.

[46] B. Misra and E. G. Sudarshan. The Zeno's paradox in quantum theory. *Journal of Mathematical Physics*, 18(4):756–763, 1977. DOI: 10.1063/1.523304.

[47] R. M. Parrish and P. L. McMahon. Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation. *arXiv preprint arXiv:1909.08925*, 2019.

[48] D. Poulin and P. Wocjan. Preparing ground states of quantum many-body systems on a quantum computer. *Phys. Rev. Lett.*, 102(13):130503, 2009. DOI: 10.1103/PhysRevLett.102.130503.

[49] D. Poulin and P. Wocjan. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. *Physical review letters*, 103 (22):220502, 2009. DOI: 10.1103/PhysRevLett.103.220502.

[50] D. Poulin, A. Kitaev, D. S. Steiger, M. B. Hastings, and M. Troyer. Quantum algorithm for spectral measurement with a lower gate count. *Physical review letters*, 121 (1):010501, 2018. DOI: 10.1103/PhysRevLett.121.010501.

[51] E. Y. Remez. Sur la détermination des polynômes d'approximation de degré donnée. *Comm. Soc. Math. Kharkov*, 10(196):41–63, 1934.

[52] Y. Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2003. DOI: 10.1137/1.9780898718003.

[53] S. Sachdeva and N. K. Vishnoi. Faster algorithms via approximation theory. *Theoretical Computer Science*, 9(2):125–210, 2013. DOI: 10.1561/0400000065.

[54] R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Physical review letters*, 101(13):130504, 2008. DOI: 10.1103/PhysRevLett.101.130504.

[55] N. H. Stair, R. Huang, and F. A. Evangelista. A multireference quantum krylov algorithm for strongly correlated electrons. *arXiv preprint arXiv:1911.05163*, 2019. DOI: 10.1021/acs.jctc.9b01125.

[56] Y. Subaşı, R. D. Somma, and D. Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.*, 122:060504, 2019. DOI: 10.1103/PhysRevLett.122.060504.

[57] M. Szegedy. Quantum speed-up of Markov chain based algorithms. In *45th Annual IEEE symposium on foundations of computer science*, pages 32–41. IEEE, 2004. DOI: 10.1109/FOCS.2004.53.

[58] E. Tang. Quantum-inspired classical algorithms for principal component analysis and supervised clustering. *arXiv preprint arXiv:1811.00414*, 2018.

[59] E. Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019. DOI: 10.1145/3313276.3316310.

[60] P. Wocjan and A. Abeyesinghe. Speedup via quantum sampling. *Physical Review A*, 78(4):042336, 2008. DOI: 10.1103/PhysRevA.78.042336.

[61] L. Wossnig, Z. Zhao, and A. Prakash. Quantum linear system algorithm for dense matrices. *Phys. Rev. Lett.*, 120(5):050502, 2018. DOI: 10.1103/PhysRevLett.120.050502.

[62] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan. Variational algorithms for linear algebra. *arXiv:1909.03898*, 2019.

[63] T. J. Yoder, G. H. Low, and I. L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501, 2014. DOI: 10.1103/PhysRevLett.113.210501.

A Block-encoding

The technique of block-encoding has been recently discussed extensively [34, 44]. 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 [21]. 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. A1 gives an $(\alpha + |c|, m+1, 0)$-block-encoding of $e^{-i\theta}(A + cI)$.
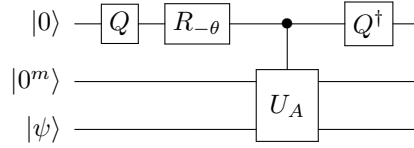

Figure A1: Quantum circuit for block-encoding of $e^{-i\theta}(A + cI)$, where $c = e^{i\theta}|c|$. $R_{-\theta} = |0\rangle \langle 0| + e^{-i\theta} |1\rangle \langle 1|$ is a phase shift gate. The three registers are the ancilla qubit for Q and $|q\rangle$, the ancilla register of U_A, and the main 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. A1 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 [34, 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 5. This is done through [34, 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.

B Implementing the reflection operator and θ-reflection operator

In this appendix we prove Theorem 4 and Corollary 5 by constructing the quantum circuits. In both the theorem and the corollary we assume, as in Theorem 3, 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 4 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 $|S_\ell(x; \delta)| \le 1$ for all $x \in [-1, 1]$. Because $-\epsilon \le \min_{y \in [-1,1]} R_\ell(y; \delta) < 0$ and $\max_{y \in [-1,1]} R_\ell(y; \delta) = 1$, we have

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

Therefore

$$-1 - 2\epsilon \le S_\ell(x; \delta) \le \frac{-1 + 2\epsilon}{1 + 2\epsilon} \le -1 + 4\epsilon, \quad x \in \mathcal{D}_\delta, \tag{A1}$$

and

$$1 - 2\epsilon \le \frac{1}{1 + 2\epsilon} \le S_\ell(0; \delta) \le 1. \tag{A2}$$

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 3. 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. (A1) and (A2) we have

$$\|S_\ell(\widetilde{H}; \widetilde{\Delta}) - R_\lambda\| \le 4\epsilon.$$

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

$$\|(\langle 0^{m+2}| \otimes I)\mathcal{U}_R(|0^{m+2}\rangle \otimes I) - R_\lambda\| = \|S_\ell(\widetilde{H}; \widetilde{\Delta}) - R_\lambda\| \le 4\epsilon.$$

Therefore $\mathcal{U}_R$ is an $(1, m+2, 4\epsilon)$-block-encoding of R_λ. Thus we have proved Theorem 4.

We then prove Corollary 5 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 A2:
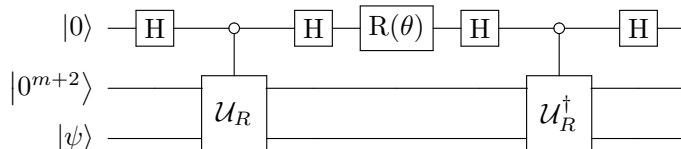

Figure A2: The quantum circuit for implementing the θ-reflection operator. H is the Hadamard gate and $\mathrm{R}(\theta) = |0\rangle\langle 0| + e^{\mathrm{i}\theta}|1\rangle\langle 1|$ is the phase-shift gate.

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 4. 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 5.

C Gate-based implementation of time-optimal adiabatic quantum computing

In Theorem 8 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 (5) and (4). It is proved in [5, 56] 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 (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 [45]. In Hamiltonian simulation, several types of input models for the Hamiltonian are in use. Hamiltonians can be input as a linear combination of unitaries [10], using its sparsity structure [1, 43], or using its block-encoding [44, 45]. For

a time-dependent Hamiltonian Low and Wiebe designed an input model based on block-encoding named HAM-T [45, 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. A3. We need to use the block-encodings U_{H_0} and U_{H_1} introduced in Appendix A, which requires $n_0 = 1$ and $n_1 = n + 4$ ancilla qubits, respectively. Our construction of HAM-T satisfies

$$(\langle s| \langle 0^{l+1+n_0}| \otimes I \otimes \langle 0^{n_1+1}|) \text{HAM-T}(|s\rangle |0^{l+1+n_0}\rangle \otimes I \otimes |0^{n_1+1}\rangle) = H(f(s))/d, \quad \text{(A3)}$$

for any $s \in \mathcal{S} = \{j/2^l : j = 0, 1, \ldots, 2^l - 1\}$.
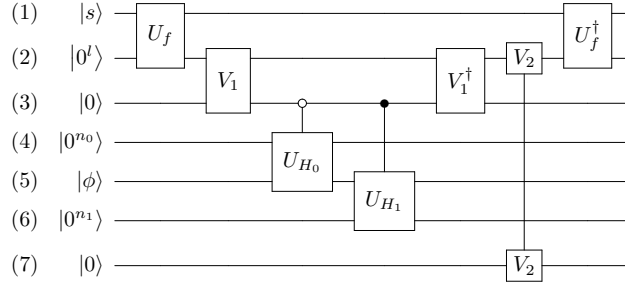

Figure A3: Quantum circuit for HAM-T. The registers from top to bottom are: (1) input register for s (2) register for storing $f(s)$ (3) register for a control qubit (4) ancilla register for U_{H_0} (5) main register for input state $|\phi\rangle$ (6) ancilla register for U_{H_1} (7) register for changing normalizing factor from $\alpha(s)$ to d.

In this unitary HAM-T we also need the unitary

$$U_f |s\rangle |z\rangle = |s\rangle |z \oplus f(s)\rangle \quad \text{(A4)}$$

to compute the scheduling function needed in the time-optimal AQC, and the unitaries

$$V_1 = \sum_{s \in \mathcal{S}} |s\rangle \langle s| \otimes \frac{1}{\sqrt{1-s+ds}} \begin{pmatrix} \sqrt{1-s} & -\sqrt{ds} \\ \sqrt{ds} & \sqrt{1-s} \end{pmatrix}$$

$$V_2 = \sum_{s \in \mathcal{S}} |s\rangle \langle s| \otimes \begin{pmatrix} \frac{\alpha(s)}{d} & -\sqrt{1-\left(\frac{\alpha(s)}{d}\right)^2} \\ \sqrt{1-\left(\frac{\alpha(s)}{d}\right)^2} & \frac{\alpha(s)}{d} \end{pmatrix}, \quad \text{(A5)}$$

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 [45].

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 D).

Following [45, 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 [5] 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 [5, 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: $\mathcal{O}(n + \log(d\kappa))$.

D   The matrix dilation method

In Theorem 8 and Theorem 11, in order to extend the time-optimal AQC method, and the QZE-based method to Hermitian indefinite matrices, we follow [5, Theorem 2], where H_0 and H_1, as constructed in Ref. [56] are given by

$$
\begin{aligned}
H_0 &= \sigma_+ \otimes [(\sigma_z \otimes I_N)Q_{+,b}] + \sigma_- \otimes [Q_{+,b}(\sigma_z \otimes I_N)], \\
H_1 &= \sigma_+ \otimes [(\sigma_x \otimes A)Q_{+,b}] + \sigma_- \otimes [Q_{+,b}(\sigma_x \otimes A)].
\end{aligned}
\tag{A6}
$$

Here $\sigma_\pm = (\sigma_x \pm i\sigma_y)/2$ and $Q_{+,b} = I_{2N} - |+\rangle |b\rangle \langle +| \langle b|$. The dimension of the dilated matrices H_0, H_1 is $4N$. The lower bound for the gap of $H(f)$ then becomes $\sqrt{(1-f)^2 + f^2/\kappa^2}$ [56]. 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 (6) by a factor of $\sqrt{2}$. The initial state is $|0\rangle |-\rangle |b\rangle$, where $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, and the goal is to obtain $|0\rangle |+\rangle |x\rangle$. In the AQC-based QLSP solver, after running the AQC we can remove the second qubit by measuring it with respect to the $\{|+\rangle, |-\rangle\}$ basis and accepting the result corresponding to $|+\rangle$. 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 $\mathfrak{A}|\mathfrak{x}\rangle = |\mathfrak{b}\rangle$ in dimension $2N$ where

$$
\mathfrak{A} = \sigma_+ \otimes A + \sigma_- \otimes A^\dagger = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}, \quad |\mathfrak{b}\rangle = |0, b\rangle.
$$

Here $\mathfrak{A}$ is a Hermitian matrix of dimension $2N$, with condition number κ and $\|\mathfrak{A}\| = 1$, and $|\mathfrak{x}\rangle = |1, x\rangle$ 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. (A6) is not needed for the eigenstate filtering step.

E   Optimality of the Chebyshev filtering polynomial

In this section we prove Lemma 2. 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 [53, Proposition 2.4], [52, Theorem 6.25], and [25, Theorem 7]:

Lemma 12. *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 $h(1) = Q_\ell(1; \Delta) - q(1)\frac{Q_\ell(y; \Delta)}{q(y)} = 1 - q(1)\frac{Q_\ell(y; \delta)}{q(y)} > 0$. $\square$

Lemma 12 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 2: 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 2, we need to use the following lemma, which directly follows from [52, Eq. (6.112)]:

Lemma 13. *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 $\pm$ 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 $T_\ell(x) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$. $\qquad\square$

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 $[-1,1]$.

F Properties of the eigenpath

In this section we construct a smooth one-parameter family of normalized quantum states $\{|x(f)\rangle\}$ satisfying Eqs. (7) and (8). $\{|0\rangle|x(f)\rangle\}$ then gives an eigenpath of the one-parameter family of Hamiltonians $\{H(f)\}$. We also prove the inequality (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

$$c'(f) = -c(f)\frac{\langle y(f)|\partial_f|y(f)\rangle}{\langle y(f)|y(f)\rangle}, \quad c(0) = 1. \tag{A7}$$

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 (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. (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 4.1. For $H(f) = (1-f)H_0 + fH_1$, where H_0 and H_1 are defined in Eqs. (5) and (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 $|w(f)\rangle = e^{i\theta(f)}|x(f)\rangle$ for some differentiable $\theta(f)$. By the geometric phase condition we can show $e^{i\theta(f)} = 1$ 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 (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

$$\|\partial_f|\bar{x}(f)\rangle\|^2 = \sum_{j:\varepsilon_j(f)\neq 0} \frac{|\langle w_j(f)|\, H'(f)|\bar{x}(f)\rangle|^2}{\varepsilon_j^2(f)}$$

$$\leq \frac{1}{\Delta_*(f)^2} \sum_{j:\varepsilon_j(f)\neq 0} |\langle w_j(f)|\, H'(f)|\bar{x}(f)\rangle|^2$$

$$\leq \frac{1}{\Delta_*(f)^2} \|H'(f)|\bar{x}(f)\rangle\|^2$$

From the definition of $H(f)$ it can be seen that $\|H'(f)\| \leq 2$. Therefore we have proved the inequality (9).

G   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 11. In Section 5.2 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 14. *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 (7) and (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 (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. (20) (21) and (15).

$$|\langle x(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2}\||x(f_{j+1})\rangle - |x(f_j)\rangle\|^2 \geq 1 - \frac{2\log^2(\kappa)}{M^2(1-1/\kappa)^2}. \qquad \text{(A8)}$$

With these tools we will first bound several overlaps in the following lemma

Lemma 15. *When $M \geq \frac{4\log^2(\kappa)}{(1-1/\kappa)^2}$ and $\epsilon_P \leq \frac{1}{128}$, we have for $j = 0, 1, \ldots, M-1$:*

(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 (A8). We then want to derive (ii) and (iii) inductively. First we have

$$|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| = |\langle \widetilde{x}(f_j)|P_0(f_j)|x(f_{j+1})\rangle + \langle \widetilde{x}(f_j)|I - P_0(f_j)|x(f_{j+1})\rangle|$$
$$\geq |\langle x(f_j)|x(f_{j+1})\rangle| \cdot |\langle x(f_j)|\widetilde{x}(f_j)\rangle| - \|(I - P_0(f_j))|\widetilde{x}(f_j)\rangle\|.$$

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

$$|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| \geq (1 - \frac{1}{2M})(1 - \nu_j) - \sqrt{1 - (1 - \nu_j)^2}$$
$$\geq 1 - \frac{1}{2M} - \nu_j - \sqrt{2\nu_j}. \qquad \text{(A9)}$$

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. (18), we have

$$|\langle \widetilde{x}(f_{j+1})|x(f_{j+1})\rangle| = \frac{|\langle \widetilde{x}(f_j)|\widetilde{P}_0(f_{j+1})|x(f_{j+1})\rangle|}{\|\widetilde{P}_0(f_{j+1})|\widetilde{x}(f_j)\rangle\|}$$
$$\geq \frac{|\langle \widetilde{x}(f_j)|P_0(f_{j+1})|x(f_{j+1})\rangle| - \epsilon_P}{\|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| + \epsilon_P}$$
$$= \frac{|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| - \epsilon_P}{|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| + \epsilon_P} \qquad \text{(A10)}$$
$$\geq 1 - \frac{2\epsilon_P}{|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle|}.$$

This leads to

$$\nu_{j+1} \leq \frac{2\epsilon_P}{|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle|} \leq \frac{2\epsilon_P}{1 - \frac{1}{2M} - \nu_j - \sqrt{2\nu_j}}, \qquad \text{(A11)}$$

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 (A11) we have

$$\nu_{j+1} \leq 4\epsilon_P,$$

which proves (ii). Taking this into (A9) we have (iii).

$\square$

An immediate corollary of (iii) in the above lemma is

$$|\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| \geq 1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P} \geq \frac{1}{2},$$ (A12)

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_{success}. We have

$$
\begin{aligned}
p_{\text{success}} &= \prod_{j=0}^{M-1} \|\widetilde{P}_0(f_{j+1})|\widetilde{x}(f_j)\rangle\|^2 \\
&\geq \left(\prod_{j=0}^{M-2} (\|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| - \epsilon_P)\right)^2 \left(\|P_0(1)|\widetilde{x}(f_{M-1})\rangle\| - \frac{\epsilon}{4}\right)^2 \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-2} (\|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| - \epsilon_P)\right)^2 \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-1} (\|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| - \epsilon_P)\right)^2 \\
&\geq \frac{1}{16}\left(\prod_{j=0}^{M-1} \|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| - M\epsilon_P\right)^2.
\end{aligned}
$$ (A13)

In the last line we have used Lemma 14. 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. (A12) 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 15, we have

$$
\begin{aligned}
\prod_{j=0}^{M-1} \|P_0(f_{j+1})|\widetilde{x}(f_j)\rangle\| &= \prod_{j=0}^{M-1} |\langle \widetilde{x}(f_j)|x(f_{j+1})\rangle| \\
&\geq \left(1 - \frac{1}{2M} - 4\epsilon_P - 2\sqrt{2\epsilon_P}\right)^M \\
&\geq \frac{1}{2} - M(4\epsilon_P + 2\sqrt{2\epsilon_P}).
\end{aligned}
$$ (A14)

In Lemma 15 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 (A13) we have

$$p_{\text{success}} \geq \frac{1}{16}\left(\frac{1}{4} - M\epsilon_P\right)^2 \geq \frac{1}{16}\left(\frac{1}{4} - \frac{1}{162M}\right)^2 \geq \frac{1}{400},$$

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.