

Spectral Gaps with Quantum Counting Queries and Oblivious State Preparation

Almudena Carrera Vazquez¹ and Aleksandros Sobczyk^{1,2}

¹IBM Research, Rüschlikon, Switzerland

²ETH Zurich, Zurich, Switzerland

Approximating the k -th spectral gap $\Delta_k = |\lambda_k - \lambda_{k+1}|$ and the corresponding midpoint $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ of an $N \times N$ Hermitian matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, is an important special case of the eigenproblem with numerous applications in science and engineering. In this work, we present a quantum algorithm which approximates these values up to additive error $\epsilon \Delta_k$ using a logarithmic number of qubits. Notably, in the QRAM model, its total complexity (queries and gates) is bounded by $O\left(\frac{N^2}{\epsilon^2 \Delta_k^2} \text{polylog}\left(N, \frac{1}{\Delta_k}, \frac{1}{\epsilon}, \frac{1}{\delta}\right)\right)$, where $\epsilon, \delta \in (0, 1)$ are the accuracy and the failure probability, respectively. For large gaps Δ_k , this provides a speed-up against the best-known complexities of classical algorithms, namely, $O\left(N^\omega \text{polylog}\left(N, \frac{1}{\Delta_k}, \frac{1}{\epsilon}\right)\right)$, where $\omega \lesssim 2.371$ is the matrix multiplication exponent. A key technical step in the analysis is the preparation of a suitable random initial state, which ultimately allows us to efficiently count the number of eigenvalues that are smaller than a threshold, while maintaining a quadratic complexity in N . In the black-box access model, we also report an $\Omega(N^2)$ query lower bound for deciding the existence of a spectral gap in a binary (albeit non-symmetric) matrix.

Almudena Carrera Vazquez: acv@zurich.ibm.com

Aleksandros Sobczyk: sobczykalek@gmail.com, the author contributed to this work exclusively while still at IBM Research and ETH Zurich. Current affiliation: Huawei Zurich Research Center.

1 Introduction

Eigenvalue problems lie in the heart of computational science, engineering, and machine learning. Given an $N \times N$ input matrix $\mathbf{A}$, the goal is to compute (a subset of) the *eigenvalues* λ_i and/or the *eigenvectors* $\mathbf{v}_i$, for $i = 1, \dots, N$, which satisfy:

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i.$$

In this work we focus on algorithms for an important special case of the eigenproblem, namely, the approximation of the so-called *spectral gap* (or *eigengap*) and the corresponding midpoint between pairs of consecutive eigenvalues of *Hermitian* matrices, which we formally define as follows.

Problem 1.1 (Spectral gap and midpoint approximation). *Let $\mathbf{H}$ be an $N \times N$ Hermitian matrix, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, and $\Delta_k := |\lambda_k - \lambda_{k+1}|$ be the k -th spectral gap and $\mu_k := \frac{\lambda_k + \lambda_{k+1}}{2}$ is the corresponding midpoint between λ_k and λ_{k+1} . Given $\mathbf{H}$, an integer $k \in [N - 1]$, and an accuracy $\epsilon \in (0, 1)$, compute two values $\tilde{\Delta}_k, \tilde{\mu}_k$ such that:*

$$\tilde{\Delta}_k \in (1 \pm \epsilon) \Delta_k, \quad \text{and} \quad \tilde{\mu}_k \in \mu_k \pm \epsilon \frac{\Delta_k}{2} \subseteq [\lambda_{k+1}, \lambda_k].$$

No prior information regarding the eigenvalues or the eigenvectors of $\mathbf{H}$ is assumed.

Problem 1.1 has numerous applications in different areas. In numerical analysis, estimating spectral gaps is a key subroutine of so-called *spectral divide-and-conquer* algorithms for diagonalizing matrices [1, 2, 3, 4]. These methods rely on finding a “splitting” point μ that does not coincide with an eigenvalue; our midpoint estimator $\mu_k = (\lambda_k + \lambda_{k+1})/2$ provides a natural and well-conditioned choice for such a split, as it maximizes the distance to the nearest eigenvalues. In data analytics, a large gap in the spectrum of a data matrix indicates a separation between the actual information and the “noise” in the data. This phenomenon is often used in Principal Component Analysis [5, 6] to identify good splitting points, or “elbows”, for dimensionality reduction, data visualization, and noise filtering. By directly targeting the k -th gap, our algorithm enables the automated detection of these elbows without requiring the explicit computation of (potentially large) subsets of singular values or a full singular value decomposition (SVD). In spectral graph theory, gaps between the eigenvalues of the graph Laplacian reveal connectivity properties, and they can be used, for example, to identify communities, [7, 8, 9, 10].

In the study of quantum phase transitions, the closing of the k -th gap, where $\lambda_{k+1} - \lambda_k \rightarrow 0$ as the system size increases, serves as a key indicator of topological or magnetic phase transitions [11]. In the context of quantum computing, the size of the spectral gap also determines the runtime of adiabatic state preparation, since the adiabatic theorem requires the evolution time to scale inversely with (a power of) the minimum spectral gap encountered [12]. Last but not least, spectral gaps are also important in electronic structure calculations, for example, to estimate the energy gap (Fermi level) that separates the occupied and unoccupied states of the underlying system [13, 14, 15, 16]. In that direction, our gap estimator could be combined with the techniques of [17] to calculate the electron density of a system at zero temperature.

The aim of this work is to provide a detailed analysis of quantum algorithms for solving Problem 1.1, aiming for provable approximation guarantees and end-to-end *worst-case* complexity bounds.

1.1 Existing algorithms and complexity bounds

Here we give a brief overview of the related algorithms literature. We first recall that, from the Abel-Ruffini theorem, computing eigenvalues of general matrices is intractable in the algebraic model; see e.g. [18]. Besides trivial cases, such as triangular matrices, algorithms for eigenvalue problems in general return approximate solutions. The computational complexity depends on several different factors, namely, the underlying model of computation, the properties of the input matrix (for example, Hermitian, sparse, structured, or low-rank), the quantities that need to be computed (subset of eigenvalues/eigenvectors), the accuracy, and, for randomized algorithms, the success probability.

In the classical (non-quantum) regime, recent works have established that both the arithmetic and boolean worst-case complexity of *diagonalization* (approximation of *all* the eigenvalues and eigenvectors), is upper bounded by $\tilde{O}(N^\omega)$ operations [3, 19, 20, 21, 22, 4]. Here ω is the matrix multiplication exponent, currently upper bounded by $\omega < 2.371339$ [23], and $\tilde{O}$ hides polylogarithmic factors in N and potentially other parameters, such as the accuracy. In terms of solving Problem 1.1 *per-se*, [20] reports a complexity of $O\left(N^\omega \text{polylog}\left(N, \frac{1}{\Delta_k}, \frac{1}{\epsilon}\right)\right)$ operations in the Boolean RAM model.

Beyond the matrix multiplication-type worst-case complexity, fast algorithms have been studied for special classes of matrices. For instance, the eigenvalues of banded symmetric matrices with k bands can be computed in $\tilde{O}(N^2 k^{\omega-2})$ time via tridiagonal reduction [20, 24]. For symmetric diagonally dominant matrices (SDD) and graph Laplacians, the smallest k eigenvalues of an $N \times N$ SDD matrix with M non-zeros can be approximated in $\tilde{O}(M + kN/\epsilon^2)$ time [25, 26]. This is faster than the aforementioned $\tilde{O}(N^\omega)$ bounds even for $k \in \Theta(N)$ and $M \in \Theta(N^2)$, albeit with increased dependence on the accuracy ϵ (polynomial instead of logarithmic). The recent work of [27] described an elegant approach specifically targeting spectral gaps of general (non-SDD) sparse symmetric matrices based on stochastic trace estimation [28, 29, 30, 31], albeit, the current worst-case complexity analysis can grow larger than $\tilde{O}(N^\omega)$.

The literature on quantum algorithms for eigenvalue problems is vast, and, to date, it keeps growing rapidly. The majority of existing works has focused on the following problems: (i) estimation of extremal eigenvalues, i.e., the ground state energy, (ii) approximating an eigenvalue, given access to the corresponding eigenvector, or (iii) preparation of quantum states that encode eigenvalues as amplitudes. Such methods include the so-called *Variational Quantum Eigensolvers* [32, 33, 34], and methods based on *Quantum Phase Estimation* (QPE) [35, 36, 37], such as [38, 39, 40, 41, 42, 43, 44].

Here we highlight recent works that can be considered for solving Problem 1.1, with explicit worst-case complexity bounds. Low and Su [39] (see e.g. Theorem 3) recently proved sharp quantum query bounds for estimating any eigenvalue of general matrices. The query bound is, $O((a\epsilon)^{-1} \log(1/\delta))$ for Hermitian matrices, where a is the block-encoding scaling factor. The algorithm assumes as input an initial state that is ϵ -close (in the Euclidean distance) to the corresponding eigenvector. Such a state is hard to obtain in the general case, as it entails to approximate an (unknown) eigenvector.

State preparation assumptions can be relaxed for the problem of estimating the smallest eigenvalues. Recent works of [38, 45, 44, 40] suggest that $1/\text{poly}(N)$ overlap between the initial state and the target bottom- k states is sufficient. If an initial state that satisfies this requirement is readily available, then a straightforward approach to solve Problem 1.1 is to directly [45, 44] to approximate all $N - k$ smallest eigenvalues, in a “brute-force” manner. As it is clearly noted in these works, however, this approach is targeted for approximating few eigenvalues close to the end of the spectrum, in this case, when $N - k$ is small. Moreover, beyond the initial state preparation assumptions, it is required that the

$N - k$ smallest eigenvalues are strictly separated by a well-defined minimum gap. For large $N - k \in \Theta(N)$, and for general Hermitian matrices, it becomes challenging to ensure or to override these assumptions and to guarantee competitive end-to-end complexity bounds in the worst-case.

End-to-end worst-case complexity bounds were proved in recent works by Apers and de Wolf [46], and by Chen, Gilyén and de Wolf [47]. The former showed how to approximate the top- k eigenvalues and eigenvectors of SDD matrices with M non-zeros in $\tilde{O}(\sqrt{MN} + kN/\epsilon^2)$ time in the QRAM model. It is faster than the $O(M + kN/\epsilon^2)$ classical SDD algorithms for small k and large M . For *general* square matrices, [47] showed how to approximate the top- k eigenvectors in $\tilde{O}(kN^{1.5+o(1)})$, using quantum versions of the power method. As also noted by the authors, however, the algorithm is less competitive for large $k \in \Theta(N)$.

As a last remark, we note that obtaining initial states which sufficiently approximate the target eigenvectors is an interesting research topic on its own. For certain applications, an initial state can be obtained by solving an “easier” or “smaller” version of the problem as a pre-processing step. As [38] notes, this is particularly interesting for quantum chemistry applications [48]. Random states are often recommended for the general case, but they can be also expensive to prepare, as we also discuss here.

1.2 Contributions and methodology

Our main contribution is an algorithm (Algorithm 2) to solve Problem 1.1. In a high-level, it consists of two phases:

- In the first phase (Alg. 1), the spectral gap Δ_k and the midpoint μ_k are computed to *constant accuracy* (i.e., a crude initial approximation).
- The second phase (Alg. 2) uses the values returned in the first phase in order to perform a refined search in the restricted domain and increase the solution accuracy.

Both phases use the following steps to locate gaps and/or eigenvalues:

1. Encode the given Hermitian matrix in a quantum circuit.
2. Split the spectrum in two parts, by shifting the diagonal elements to a carefully chosen point.
3. Compute the spectral projector on each part of the spectrum with a polynomial approximation of the sign function.
4. Compute the trace of the projector.
5. Repeat 2-4 as a search mechanism (“counting queries”) over a grid to locate eigenvalues and/or gaps.

For Step 1, we use the block-encoding framework [49, 50], which has recently received significant attention for matrix problems. We use the following definition.

Definition 1.1 (Block-Encoding). *Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ be an s -qubit matrix, $a \geq \|\mathbf{A}\|$, $\epsilon > 0$, and $b \in \mathbb{N}$. An $(s+b)$ -Qubit Unitary $\mathbf{U}_{\mathbf{A}}$ is an (a, b, ϵ) -Block-Encoding of $\mathbf{A}$, which is also abbreviated as (a, b, ϵ) -QUBE for short, if*

$$\left\| \mathbf{A} - a\tilde{\mathbf{A}} \right\| \leq \epsilon, \quad (1)$$

where $\tilde{\mathbf{A}} := (|0\rangle^{\otimes b} \otimes \mathbf{I})\mathbf{U}_{\mathbf{A}}(|0\rangle^{\otimes b} \otimes \mathbf{I})$.

Our main result is summarized in the following theorem.

Theorem 1.1 (Informal, see Theorem 3.1). *Let $\mathbf{H}$ be an $N \times N$ Hermitian matrix with eigenvalues $-1/2 \leq \lambda_N \leq \lambda_{N-1} \leq \dots \leq \lambda_1 \leq 1/2$, $\Delta_k := |\lambda_k - \lambda_{k+1}|$, and $\mu_k := \frac{\lambda_k + \lambda_{k+1}}{2}$, and let $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$ denote the cost (gate complexity) of the unitary $\mathbf{U}_{\mathbf{H}}$ which forms an $(a, b, \epsilon_{\text{enc}})$ -block-encoding of $\mathbf{H}$. The proposed Algorithm 2 takes as input $\mathbf{H}$, an integer $k \in [N-1]$, an accuracy $\epsilon \in (0, 1)$, and a failure probability parameter $\delta \in (0, 1/2)$, and it returns values $\tilde{\lambda}_{k+1}$, $\tilde{\lambda}_{k+1}$, $\tilde{\mu}_k$, and $\tilde{\Delta}_k$, which satisfy:*

$$\tilde{\lambda}_{k+1} \in [\lambda_{k+1}, \lambda_{k+1} + \epsilon \frac{\Delta_k}{2}], \quad \tilde{\lambda}_k \in [\lambda_k - \epsilon \frac{\Delta_k}{2}, \lambda_k], \quad \tilde{\mu}_k \in \mu_k \pm \epsilon \frac{\Delta_k}{2}, \quad \tilde{\Delta}_k \in (1 \pm \epsilon) \Delta_k.$$

The total complexity, including all queries to unitaries and auxiliary gates, is bounded by:

$$O\left(\left[N^2 + N \cdot \frac{a^2 b \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})}{\epsilon^2 \Delta_k^2}\right] \text{polylog}\left(N, a, \epsilon^{-1}, \Delta_k^{-1}, \delta^{-1}\right)\right), \quad (2)$$

where $\epsilon_{\text{enc}} \in \Theta\left(\frac{\epsilon^2 \Delta_k^2}{a^2 N}\right)$.

Notably, in the QRAM model, the complexity becomes $\tilde{O}(N^2 \Delta_k^{-2} \epsilon^{-2})$. This is faster than the $\tilde{O}(N^\omega)$ arithmetic complexity of the best-known classical algorithms, for all matrices with sufficiently large gaps. This provides evidence that quantum algorithms can beat “matrix multiplication-time” for these problems, at least for the current bound of ω .

While the general idea is simple, there are several non-trivial parts that need to be addressed to obtain competitive end-to-end complexities. Two major challenges, which can be of independent interest, were the choice of the splitting points, and the preparation of appropriate initial states, which we discuss further below.

Informed choice of splitting points via condition number estimation

As mentioned, the splitting points to split the spectrum and count eigenvalues. If the chosen point is close to an actual eigenvalue, the condition number of the shifted matrix can grow arbitrarily, and it becomes prohibitively expensive to approximate the corresponding spectral projector. Indeed, the convergence of iterative methods, such as the Newton-Schultz iteration or other rational approximations [51], scales as $\Omega(1/\Delta)$, where Δ is the distance of the splitting point from the closest eigenvalue [49]. For this reason, many existing algorithms that follow this approach require prior knowledge of Δ (or at least a suitable lower bound), which is generally not trivial to obtain.

To avoid this problem, we use a dedicated subroutine which tests whether the gap Δ is larger than a well-defined threshold, before executing any counting query. This is achieved by conveniently applying the ground-energy estimation algorithm of [38] in order to estimate the smallest singular value of the matrix, after shifting its diagonal elements to the chosen splitting point. If this singular value is “large enough”, it means that we can safely count eigenvalues. If not, we keep searching for smaller gaps. This way we have full control of the value of Δ , and, consequently, of the cost of eigenvalue counting.

Initial state preparation

In order to execute our singular value estimation subroutine efficiently, we need to address a second major challenge, which involves the preparation of an appropriate initial state. The algorithm of [38] requires as input an initial state $|x\rangle$ which will overlap appropriately

with the corresponding singular vector. In particular, this target singular vector can be any fixed state $|y\rangle$, and we need to ensure that the initial guess $|x\rangle$ will satisfy

$$|\langle x|y\rangle| \geq \frac{1}{N^{O(1)}}. \quad (3)$$

As the authors of [38, 45] note, such an initial state can be efficiently prepared for certain classes of problems. For instance, if $|y\rangle$ is a binary string with exactly one nonzero entry, a Hadamard state is sufficient, or, in quantum chemistry applications, the Hartree-Fock state can be pre-computed as a “cheap” initial approximation to the ground state [48].

In the general case, where $|y\rangle$ can be any arbitrary state without any explicit prior information, the analysis becomes more involved. In the context of QPE, a common recommendation is to use a random state. However, if the random distribution is not chosen carefully, it can be harmful rather than beneficial. Contributing to this direction, in Section 2.1 we analyze a randomized state preparation method, which achieves all the desired properties. It is *oblivious* to the target state, in a sense that it guarantees the desired overlap $1/N$, for any fixed target state, with probability at least $3/5$. Our construction is an ensemble of a random Rademacher state, together with a randomized Hadamard transform [52, 53, 54]. The latter is a well-established tool in Randomized Numerical Linear Algebra, and it is crucial for our analysis due to its “flattening” properties, commonly attributed as a consequence of the Heisenberg principle.

The size of the final quantum circuit scales as $O(N)$, and it requires a total of $O(N)$ random classical bits to prepare. While this cost is prohibitive for applications that target exponential quantum speed-ups, it is “sufficiently cheap” for our analysis, and it can be useful for other works that target polynomial quantum speed-ups.

Lower bounds

The aforementioned upper bounds are also supported with a $\Omega(N^2)$ query lower bound for the problem of deciding whether there exists a gap between two consecutive eigenvalues of a (non-symmetric) binary matrix in the black-box access model. The techniques used here are based on “majority” arguments, inspired by [55, 56, 57]. Evidently, these lower bounds are not tight, as they do not match the reported upper bound. In order to close this complexity gap, we think that one should use entirely different techniques, both for the upper and the lower bounds. This is left as a major open question.

1.3 Notation

$\mathcal{H}_S$ denotes an explicit Hilbert space and $\mathcal{H}_A$ an ancillary space for post-selection. When we want to emphasize its dimension d , we write $\mathcal{H}_A \cong \mathbb{C}^d$. The total Hilbert space is given by $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_A$. Matrices and vectors are denoted with bold capital and small letters, respectively. For a matrix $\mathbf{A}$, $\mathbf{A}^\dagger$ is the conjugate transpose, $\|\mathbf{A}\|$ denotes its spectral norm, $\|\mathbf{A}\|_F$ its Frobenius norm, and $\Lambda(\mathbf{A})$ denotes its spectrum. $\mathcal{C}(\mathbf{U})$ denotes the gate complexity of the unitary $\mathbf{U}$, and we will write $\mathcal{C}(\mathbf{U}; \epsilon, \delta)$ when the complexity of $\mathbf{U}$ is parametrized by ϵ, δ .

1.4 Model of computation

The complexity of a quantum algorithm is two-fold. We define the *query complexity* to be the total number of calls to the block-encoding of the original matrix. The *gate complexity* is the total number of elementary one- and two-qubit quantum gates, i.e., the size of the

quantum circuit. For all algorithms, we explicitly mention both the query and the gate complexities, taking into account all the involved parameters, such as the matrix size, the accuracy, and the success probabilities. For our state-preparation analysis, we also assume access to a (classical) random number generator that produces independent, uniformly random bits.

In Section 4, we specialize the analysis in the so-called Quantum-RAM (or QRAM) model [58]. In this model the quantum processor has access to a memory that stores classical bits. These bits can be read in superposition, but we can only write zeros and ones, just as in the classical case. This model is not necessary for our main analysis, but we dedicate Section 4 to it since it is used in theoretical works, e.g., for graph and matrix computations [46, 47, 41]. For a discussion on QRAM architectures we refer to [59, 60].

1.5 Outline

Section 2 contains the analysis of the main subroutines that are used by the main algorithms, namely, for state preparation, computing the minimum singular value, and counting eigenvalues that are smaller than a threshold. Section 3 contains the main algorithm for computing spectral gaps and eigenvalue pairs. In Section 4 we report the analysis in the QRAM model. In Section 5 we prove a query lower bound for deciding the existence of a spectral gap in binary matrices in the black-box access model. We finally provide some concluding remarks in Section 6.

Acknowledgements

We are grateful to Ivano Tavernelli and Simon Apers for valuable discussions.

2 Main Subroutines

Here we describe the main subroutines that are used as building blocks for our main algorithms. Some of the techniques reported here rely on existing results on block-encodings and quantum eigenvalue transformation from [49, 61, 38, 50], which are summarized in Appendix A.

2.1 Oblivious State Preparation

In many applications of QPE, it is required to prepare an initial state $|x\rangle$ that is not orthogonal to the target state $|y\rangle$ (e.g. the ground state). A sufficient condition is typically $|\langle x|y\rangle|^2 \geq \Omega(1/N)$. The state $|x\rangle = |+\rangle := \mathbf{H}|0\rangle$ is sufficient for many applications, e.g., when the desired state is a binary string. For arbitrary $|x\rangle$, however, there is no theoretical guarantee that $|+\rangle$ is not orthogonal to $|x\rangle$. For example, it is known that $|+\rangle$ spans the kernel of a connected graph Laplacian, and it is therefore orthogonal to *every* eigenvector that corresponds to a nonzero eigenvalue.

A reasonable approach to circumvent this limitation is to use random states. However, as already mentioned previously, this needs to be handled with care. Here we describe a random state preparation, inspired by well-known techniques from randomized matrix algorithms [52]. Before stating the main result, we define a state-preparation unitary.

Definition 2.1. *A distribution Π of random quantum circuits forms a (γ, δ) -Oblivious State Preparation, or (γ, δ) -OSP, if the unitary $\mathbf{U}_S \sim \Pi$ operates on $O(\log(N))$ qubits*

and it prepares a state $|x\rangle = \mathbf{U}_S |0\rangle$ such that, for any fixed state $|y\rangle \in \mathbb{C}^N$:

$$\Pr[|\langle x|y\rangle| < \gamma] \leq \delta.$$

Lemma 2.1. Let $|y\rangle \in \mathbb{C}^N$ be any fixed state (e.g. the target ground-state). Let $|x\rangle = \mathbf{DH}|z\rangle$. Here $\mathbf{D}$ is diagonal with independent $\{\pm 1\}$ diagonal elements, $\mathbf{H} := \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}^{\otimes \log(N)}$ is a Hadamard unitary, and $|z\rangle \in \left\{\pm \frac{1}{\sqrt{N}}\right\}^N$ has independent Rademacher elements. Let $\mathbf{Z}$ be a unitary such that $\mathbf{Z}|0\rangle = |z\rangle$. Then, for all $N \geq C$, where $C \approx 22$ is a fixed constant, the distribution of unitaries $\mathbf{U}_S = \mathbf{DHZ}$ is an $(\frac{1}{2N}, \frac{2}{5})$ -OSP, that is,

$$\Pr\left[|\langle x|y\rangle| > \frac{1}{2N}\right] \geq \frac{3}{5},$$

The state $|x\rangle$ can be prepared with $O(N)$ elementary gates, i.e., $\mathcal{C}(\mathbf{U}_S) \in O(N)$.

Proof. Let us denote $\mathbf{x} = |x\rangle$, $\mathbf{z} = |z\rangle$, and $\mathbf{y} = |y\rangle$ to simplify the notation. The operator $\mathbf{HD}$ is a well-studied tool in randomized linear algebra (see e.g. [52]), which has the property that it “flattens” the elements of any fixed vector with high probability. Indeed, using Hoeffding’s inequality, we can see that for any index $i \in [N]$ it holds that

$$\Pr[|(\mathbf{HD}\mathbf{y})_i| \geq t] \leq 2 \exp(-Nt^2/2).$$

Taking a union bound for all $i \in [N]$ and setting $t = \sqrt{\frac{2 \ln(2N/\delta)}{N}}$, we can conclude that

$$\Pr[\|\mathbf{HD}\mathbf{y}\|_\infty \geq t] \leq \delta.$$

Denote $\mathbf{v} := \mathbf{HD}\mathbf{y}$. It remains to get a lower bound on the overlap $|\mathbf{x}^\top \mathbf{y}| = |\mathbf{z}^\top \mathbf{v}|$. Since $\mathbf{HD}$ is unitary, $\mathbf{v}$ is a unit vector with flattened entries, which is the key observation to obtain the desired bound. Let $\mathbf{v}_m$ and $\mathbf{v}_{m'}$ be the largest and second largest elements of $\mathbf{v}$ in modulus, where $m, m' \in [N]$. It holds that $|\mathbf{v}_m|^2 \in [1/N, t^2]$. For $\mathbf{v}_{m'}$, observe that

$$|\mathbf{v}_{m'}|^2 \geq \frac{\|\mathbf{v}\|_2^2 - |\mathbf{v}_m|^2}{N-1} \geq \frac{1-t^2}{N-1}.$$

For $\delta = 1/5$ and $N \gtrsim 22$, we have that $|\mathbf{v}_{m'}|^2 \geq 1/(2N-2) > 1/(2N)$. We now condition on the success of the flattening, and proceed to get a bound for the overlap. Let $\mathbf{z}_A, \mathbf{z}_B, \mathbf{z}_C, \mathbf{z}_D$ be four vectors that have exactly the same signs at all the positions except $\{m, m'\}$, where in these two positions they have the signs $\{+1, +1\}, \{+1, -1\}, \{-1, +1\}, \{-1, -1\}$, respectively. The following relations hold:

$$\begin{aligned} |\mathbf{z}_A^\top \mathbf{v} - \mathbf{z}_B^\top \mathbf{v}| &\geq \frac{2|\mathbf{v}_{m'}|}{\sqrt{N}} &\implies \max\{|\mathbf{z}_A^\top \mathbf{v}|, |\mathbf{z}_B^\top \mathbf{v}|\} &\geq \frac{|\mathbf{v}_{m'}|}{\sqrt{N}} > \frac{1}{2N} \\ |\mathbf{z}_A^\top \mathbf{v} - \mathbf{z}_C^\top \mathbf{v}| &\geq \frac{2|\mathbf{v}_m|}{\sqrt{N}} &\implies \max\{|\mathbf{z}_A^\top \mathbf{v}|, |\mathbf{z}_C^\top \mathbf{v}|\} &\geq \frac{|\mathbf{v}_m|}{\sqrt{N}} > \frac{1}{2N} \\ |\mathbf{z}_A^\top \mathbf{v} - \mathbf{z}_D^\top \mathbf{v}| &\geq \frac{2(|\mathbf{v}_m| + |\mathbf{v}_{m'}|)}{\sqrt{N}} &\implies \max\{|\mathbf{z}_A^\top \mathbf{v}|, |\mathbf{z}_D^\top \mathbf{v}|\} &\geq \frac{|\mathbf{v}_m| + |\mathbf{v}_{m'}|}{\sqrt{N}} > \frac{1}{2N} \\ |\mathbf{z}_B^\top \mathbf{v} - \mathbf{z}_C^\top \mathbf{v}| &\geq \frac{2(|\mathbf{v}_m| + |\mathbf{v}_{m'}|)}{\sqrt{N}} &\implies \max\{|\mathbf{z}_B^\top \mathbf{v}|, |\mathbf{z}_C^\top \mathbf{v}|\} &\geq \frac{|\mathbf{v}_m| + |\mathbf{v}_{m'}|}{\sqrt{N}} > \frac{1}{2N} \\ |\mathbf{z}_B^\top \mathbf{v} - \mathbf{z}_D^\top \mathbf{v}| &\geq \frac{2|\mathbf{v}_m|}{\sqrt{N}} &\implies \max\{|\mathbf{z}_B^\top \mathbf{v}|, |\mathbf{z}_D^\top \mathbf{v}|\} &\geq \frac{|\mathbf{v}_m|}{\sqrt{N}} > \frac{1}{2N} \end{aligned}$$

$$|\mathbf{z}_C^\top \mathbf{v} - \mathbf{z}_D^\top \mathbf{v}| \geq \frac{2|\mathbf{v}_{m'}|}{\sqrt{N}} \implies \max \left\{ |\mathbf{z}_C^\top \mathbf{v}|, |\mathbf{z}_D^\top \mathbf{v}| \right\} \geq \frac{|\mathbf{v}_{m'}|}{\sqrt{N}} > \frac{1}{2N}.$$

Now, denote $L := 1/(2N)$, and let A be the event that $|\mathbf{z}_A^\top \mathbf{v}| > L$, $B : |\mathbf{z}_B^\top \mathbf{v}| > L$, $C : |\mathbf{z}_C^\top \mathbf{v}| > L$, and $D : |\mathbf{z}_D^\top \mathbf{v}| > L$. From the relations above, we can see that (deterministically), we are in one of the following five cases:

$$\begin{aligned} A \wedge B \wedge C \wedge D \\ \neg A \implies B \wedge C \wedge D \\ \neg B \implies A \wedge C \wedge D \\ \neg C \implies A \wedge B \wedge D \\ \neg D \implies A \wedge B \wedge C. \end{aligned}$$

In all cases, at least three out of four vectors $\{\mathbf{z}_A, \mathbf{z}_B, \mathbf{z}_C, \mathbf{z}_D\}$ satisfy the desired lower bound (deterministically). Then, since we pick any of the four vectors uniformly at random, the success probability is at least $3/4$.

Since the flattening succeeds with probability $4/5$, both events succeed with probability at least $\frac{3}{4} \cdot \frac{4}{5} = \frac{3}{5}$, which concludes the proof. $\square$

2.2 Computing the smallest singular value

Here we describe the subroutine which computes the smallest singular value of a block-encoded matrix. [38] gave an algorithm with near optimal dependence on the precision ϵ to find the smallest eigenvalue of a Hermitian matrix $\mathbf{H}$.

Theorem 2.1 (Ground energy, Imported Thm. 8 from [38]). *Let $\mathbf{H} = \sum_k \lambda_k |\psi_k\rangle \langle \psi_k| \in \mathbb{C}^{N \times N}$, where $\lambda_k \geq \lambda_{k+1}$, be a Hermitian matrix given through its $(\alpha, m, 0)$ -block-encoding $\mathbf{U}_\mathbf{H}$. Let $|\phi_0\rangle$ be an initial state prepared by circuit $\mathbf{U}_S$, with the promise $|\langle \phi_0 | \psi_0 \rangle| \geq \gamma$. Then the ground energy λ_N can be estimated to additive precision ϵ with probability $1 - \delta$ with the following costs:*

1. Queries to $\mathbf{U}_\mathbf{H}$: $O\left(\frac{\alpha}{\gamma\epsilon} \log\left(\frac{\alpha}{\epsilon}\right) \log\left(\frac{1}{\gamma}\right) \log\left(\frac{\log(\alpha/\epsilon)}{\delta}\right)\right)$,
2. Queries to $\mathbf{U}_S$: $O\left(\frac{1}{\gamma} \log\left(\frac{\alpha}{\epsilon}\right) \log\left(\frac{\log(\alpha/\epsilon)}{\delta}\right)\right)$,
3. Number of qubits: $O\left(\log(N) + m + \log\left(\frac{1}{\gamma}\right)\right)$,
4. Other elementary gates: $O\left(\frac{m\alpha}{\gamma\epsilon} \log\left(\frac{\alpha}{\epsilon}\right) \log\left(\frac{1}{\gamma}\right) \log\left(\frac{\log(\alpha/\epsilon)}{\delta}\right)\right)$.

This can be applied appropriately on the Gramian $\mathbf{H}^\dagger \mathbf{H}$ to compute the minimum singular value.

Lemma 2.2 (QSMIN). *Let $\mathbf{H} \in \mathbb{C}^{N \times N}$ be a Hermitian matrix, with singular values $0 \leq \sigma_N \leq \sigma_{N-1} \leq \dots \leq \sigma_1 \leq 1$. Let $\epsilon_\sigma \in (0, 1/2]$ be an accuracy parameter, and assume that we have access to:*

- $\mathbf{U}_\mathbf{H}$, an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}$, $\epsilon_{\text{enc}} \in [0, \epsilon_\sigma/4a]$, with cost $\mathcal{C}(\mathbf{U}_\mathbf{H}; \epsilon_{\text{enc}})$
- $\mathbf{\Pi}$, a $(\gamma, \delta_{\text{osp}})$ -OSP, where $\delta_{\text{osp}} \in [0, 1/2)$, and $\mathbf{U}_S \sim \mathbf{\Pi}$ has cost $\mathcal{C}(\mathbf{U}_S)$.

We can compute a boolean string $X \leftarrow \text{QSMIN}(\mathbf{U}_\mathbf{H}, \epsilon_\sigma, \delta_\sigma)$, such that $\tilde{\sigma}^2 := 0.X \in [0, 1]$ satisfies $|\tilde{\sigma}^2 - \sigma_N^2| \leq \epsilon_\sigma$ with probability at least $1 - \delta_\sigma$, with the following costs:

1. *Queries to $\mathbf{U}_H$* : $O\left(\frac{a^2}{\gamma\epsilon_\sigma} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/\delta_\sigma)}{1-2\delta_{\text{osp}}}\right)$,
2. *Queries to random circuits $\mathbf{U}_S \sim \Pi$* : $O\left(\frac{1}{\gamma} \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/\delta_\sigma)}{1-2\delta_{\text{osp}}}\right)$,
3. *Number of qubits*: $O\left(\log(N) + b + \log\left(\frac{1}{\gamma}\right)\right)$,
4. *Other elementary gates*: $O\left(\frac{a^2b}{\gamma\epsilon_\sigma} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/\delta_\sigma)}{1-2\delta_{\text{osp}}}\right)$.

Proof. Given $\mathbf{U}_H$, an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}$ with cost $\mathcal{C}(\mathbf{U}_H; \epsilon_{\text{enc}})$, it holds that $\mathbf{U}_H^\dagger$ is an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}^\dagger$, also with cost $\mathcal{C}(\mathbf{U}_H; \epsilon_{\text{enc}})$. We can combine these two block-encodings as in [50, Lemma 4] to create an $(a^2, 2b, 2a\epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}^\dagger\mathbf{H}$, denoted by $\mathbf{U}_{H^\dagger H}$, with cost equal to $\mathcal{C}(\mathbf{U}_{H^\dagger H}; \epsilon_{\text{enc}}) = 2\mathcal{C}(\mathbf{U}_H; \epsilon_{\text{enc}})$. The minimum eigenvalue of $\mathbf{H}^\dagger\mathbf{H}$ is the square of the minimum singular value of $\mathbf{H}$.

The following steps are performed. We first prepare a state $|\psi\rangle = \mathbf{U}_S |0\rangle$ where $\mathbf{U}_S \sim \Pi$. Then, we use Theorem 2.1 with inputs $|\psi\rangle$, $\mathbf{U}_{H^\dagger H}$, $\epsilon = \epsilon_\sigma/2$, and $\delta = 1/4 - \delta_{\text{osp}}/2 \in (0, 1/4]$. The algorithm returns a bit-string $\tilde{\sigma}^2 \in [0, 1]$ which satisfies $|\tilde{\sigma}^2 - \sigma_N(\widetilde{\mathbf{H}^\dagger\mathbf{H}})| \leq \epsilon_\sigma/2$ with probability at least:

$$1 - \delta_{\text{osp}} - \delta = 3/4 - \delta_{\text{osp}}/2 \in (1/2, 3/4].$$

Here $\widetilde{\mathbf{H}^\dagger\mathbf{H}}$ is the block-encoded version of $\mathbf{H}^\dagger\mathbf{H}$, which satisfies $\|\widetilde{\mathbf{H}^\dagger\mathbf{H}} - a^2\mathbf{H}^\dagger\mathbf{H}\| \leq 2a\epsilon_H$.

From the stability of singular values, which is a consequence of Weyl's inequality, we conclude that $|\tilde{\sigma}^2 - \sigma_N^2(\mathbf{H})| \leq 2a\epsilon_H + \epsilon_\sigma/2 \leq \epsilon_\sigma$, due to the assumption that $\epsilon_{\text{enc}} \leq \epsilon_\sigma/4a$. From Theorem 2.1, the cost of such a run is bounded by

- $O\left(\frac{a^2}{\gamma\epsilon_\sigma} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right)\right)$ queries to $\mathbf{U}_{H^\dagger H}$,
- $O\left(\frac{1}{\gamma} \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right)\right)$ queries to $\mathbf{U}_S$,
- $O\left(\log(N) + b + \log\left(\frac{1}{\gamma}\right)\right)$ qubits,
- $O\left(\frac{a^2b}{\gamma\epsilon_\sigma} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon_\sigma}\right) \log\left(\frac{\log(a/\epsilon_\sigma)}{1-2\delta_{\text{osp}}}\right)\right)$ other elementary gates.

To boost the success probability we can perform multiple runs and keep the median (often referred to as the “median trick”). Note that the returned value $\tilde{\sigma}^2$ is always in $[0, 1]$, regardless whether the algorithm succeeds or not. Now, assume that we execute r runs, and consider the following random variables for each run:

$$X_i = \begin{cases} 1, & \text{if } |\tilde{\sigma}^2 - \sigma_N^2| \leq \epsilon_\sigma, \\ 0, & \text{else,} \end{cases}$$

where $i = 1, \dots, r$. X_i are independent Bernoulli variables which succeed with probability p , where $p \geq 3/4 - \delta_{\text{osp}}/2$. Let $X = \sum_{i=1}^r X_i$. We need to bound the probability that $X > r/2$. It holds that $\mathbb{E}[X] = rp$. Since, by assumption, $\delta_{\text{osp}} < 1/2$, we can use a multiplicative Chernoff bound (see e.g. [62]), which gives $\Pr[X < (1 - \eta)rp] < \exp(-\eta rp)$. For $\eta = 1 - \frac{1}{2p}$, we have that

$$\Pr[X \leq r/2] < \exp(-r(p - 1/2)) \leq \exp(-r(1 - 2\delta_{\text{osp}})/4).$$

Setting $r = O\left(\frac{\log(1/\delta)}{1-2\delta_{\text{osp}}}\right)$ we obtain the desired success probability $1 - \delta$. $\square$

2.3 Counting eigenvalues

In this section we describe the second subroutine of our algorithm, which is used to count the eigenvalues of a matrix that are smaller than a given threshold. This is a quantum variant of a standard subroutine in spectral divide-and-conquer eigensolvers [63, 64, 2, 3, 21, 19], often referred to as “NEGCOUNT.” Formally, we have the following lemma.

Lemma 2.3 (QCOUNT). *Let $\mathbf{A}$ be a Hermitian matrix with $\|\mathbf{A}\| \leq 1$, and a promise that $\sigma_{\min}(\mathbf{A}) \geq \Delta$, for some $\Delta > 0$. Given $\mathbf{U}_{\mathbf{A}}$, an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{A}$, if $\epsilon_{\text{enc}} \leq \frac{\Delta^2}{32a^2C_{\text{sgn}}}N$ for some absolute constant C_{sgn} , then there exists a quantum algorithm which returns $z \leftarrow \text{QCOUNT}(\mathbf{U}_{\mathbf{A}}, \Delta, \delta)$, the true number of eigenvalues of $\mathbf{A}$ that are smaller than zero, with probability at least $1 - \delta$, with a cost:*

1. Queries to $\mathbf{U}_{\mathbf{A}}$: $O\left(\frac{a^2N}{\Delta} \log(N) \log\left(\frac{1}{\delta}\right)\right)$,
2. Number of qubits: $O(\log(N) + b)$,
3. Other elementary gates: $O\left(b\frac{a^2N}{\Delta} \log(N) \log\left(\frac{1}{\delta}\right)\right)$,
4. Total cost: $O\left((\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon_{\text{enc}}) + b) \cdot \frac{a^2N}{\Delta} \log(N) \log\left(\frac{1}{\delta}\right)\right)$.

Proof. The following steps achieve the result.

1. We use Corollary A.1 to prepare a block-encoding for $\text{sgn}(\mathbf{A})$. As long as $\epsilon_{\text{enc}} \leq \epsilon_{\text{sgn}} \frac{\Delta^2}{C_{\text{sgn}}} a^2$ for some absolute constant C_{sgn} , the corollary returns an $(1, b + 2, \epsilon_{\text{sgn}})$ -QUBE $\mathbf{U}_{\text{sgn}}$ for $\text{sgn}(\mathbf{A})$ with cost

$$\mathcal{C}(\mathbf{U}_{\text{sgn}}; \Delta, \epsilon_{\text{enc}}) = O\left((\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon_{\text{enc}}) + b) \cdot \frac{a}{\Delta} \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right)\right).$$

2. Thereafter, Corollary A.2 is applied to compute the trace of $\text{sgn}(\mathbf{A})$ with accuracy ϵ_{tr} , which takes time

$$\begin{aligned} O\left(\mathcal{C}(\mathbf{U}_{\text{sgn}}; \Delta, \epsilon_{\text{enc}}) \frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta}\right)\right) &= O\left((\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon_{\text{enc}}) + b) \frac{a}{\Delta} \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right) \frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta}\right)\right) \\ &= O\left((\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon_{\text{enc}}) + b) \frac{a^2N}{\Delta \epsilon_{\text{tr}}} \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right) \log\left(\frac{1}{\delta}\right)\right). \end{aligned} \quad (4)$$

3. We now set $\epsilon_{\text{sgn}} = 1/(8N)$, $\epsilon_{\text{tr}} = 1/4$, and $\epsilon_{\text{enc}} = \epsilon_{\text{sgn}} \frac{\Delta^2}{C_{\text{sgn}} a^2} = \frac{\Delta^2}{8a^2 C_{\text{sgn}} N}$. Note that this value of ϵ_{enc} satisfies all the requirements of the previous steps. Plugging-in the parameters, the total complexity becomes

$$O\left((\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon_{\text{enc}}) + b) \cdot \frac{a^2N}{\Delta} \log(N) \log\left(\frac{1}{\delta}\right)\right).$$

It remains to verify the correctness. Since $\epsilon_{\text{enc}} < \Delta/2$, the number of eigenvalues of $\mathbf{A}$ (and also $\tilde{\mathbf{A}}$) that are smaller than zero, is equal to $z = \frac{1}{2} \text{tr}(\mathbf{I} - \text{sgn}(\tilde{\mathbf{A}})) = \frac{N}{2} - \frac{1}{2} \text{tr}(\text{sgn}(\tilde{\mathbf{A}})) = \frac{N}{2} - \frac{1}{2} \text{tr}(\text{sgn}(\mathbf{A}))$.

From Corollary A.2, the above steps return a value $\tilde{t}$ which satisfies $|\tilde{t} - \text{tr}(\text{sgn}(\mathbf{A}))| \leq 1/4$. This means that the value $\tilde{z} = N/2 - \tilde{t}/2$ satisfies $|z - \tilde{z}| \leq 1/8$. Therefore, we can round $\tilde{z}$ to the nearest integer to recover the true z . $\square$

3 Main algorithms

We now have all the required tools to describe our main algorithms for the k -th spectral gap and midpoint, namely, $\Delta_k = \lambda_k - \lambda_{k+1}$ and $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$. Two algorithms are described. The first one, Algorithm 1 takes as input a Hermitian matrix $\mathbf{H}$ (scaled to have $\|\mathbf{H}\| \leq 1/2$), an integer $k \in [N - 1]$, and a failure probability parameter δ . It returns constant factor approximations to Δ_k and μ_k .

The second Algorithm 2, takes as input $\mathbf{H}$, k , δ , and an accuracy parameter ϵ . It first uses Algorithm 1 as a subroutine to compute a coarse (constant factor) approximation to Δ_k and μ_k , and then uses these approximations to refine the accuracy of the solution to the given ϵ . As a side-result, it also returns ϵ -approximations to the actual eigenvalues λ_k and λ_{k+1} around the gap.

The complexity of these algorithms depends on the value of Δ_k . However, we highlight that no prior knowledge or lower bound on Δ_k is required. The only required parameters are the ones given as input to the algorithms. All other parameters that are involved in the complexity are computed/approximated on-the-fly, if required.

Lemma 3.1 (Alg. 1, QGAPconst). *Let $\mathbf{H}$ be an $N \times N$ Hermitian matrix with eigenvalues*

$$1/2 \geq \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N \geq -1/2.$$

Let:

- $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$ be the cost of the unitary $\mathbf{U}_{\mathbf{H}}$ which forms an $(a, b, \epsilon_{\text{enc}})$ -QUBE of $\mathbf{H}$.
- Π be a $(\gamma, \delta_{\text{osp}})$ -OSP distribution, where $\mathbf{U}_S \sim \Pi$ has cost $\mathcal{C}(\mathbf{U}_S)$.

Then, given $\mathbf{H}$ and an integer $k \in [N - 1]$, Algorithm 1 (QGAPconst($\mathbf{H}, k, \delta$)) computes an approximate midpoint $\tilde{\mu}_k$ and gap $\tilde{\Delta}_k$, such that:

$$|\tilde{\mu}_k - \mu_k| \leq \frac{7}{16} \Delta_k, \quad \tilde{\Delta}_k \in [\Delta_k/2, 4\Delta_k].$$

The algorithm succeeds with probability at least $1 - \delta$. It makes at most:

1. *Queries to unitaries with gate complexity $O(\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}}))$, where $\epsilon_{\text{enc}} \in \Theta\left(\frac{\Delta_k^2}{a}\right)$:*

$$Q_1 := O\left(\frac{a^2}{\gamma \Delta_k^2 (1 - 2\delta_{\text{osp}})} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\Delta_k}\right) \log\left(\frac{\log(a/\Delta_k)}{1 - 2\delta_{\text{osp}}}\right) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\Delta_k}\right)\right),$$

2. *Queries to $\mathbf{U}_S$:*

$$O\left(\frac{1}{\gamma(1 - 2\delta_{\text{osp}})} \log\left(\frac{a}{\Delta_k}\right) \log\left(\frac{\log(a/\Delta_k)}{1 - 2\delta_{\text{osp}}}\right) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\Delta_k}\right)\right),$$

3. *Queries to unitaries with gate complexity $O(\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon'_{\text{enc}}))$, where $\epsilon'_{\text{enc}} \in \Theta\left(\frac{\Delta_k^2}{a^2 N}\right)$:*

$$Q_2 := O\left(\frac{a^2 N}{\Delta_k} \log(N) \log\left(\frac{1}{\Delta_k \delta}\right) \log\left(\frac{1}{\Delta_k}\right)\right),$$

4. *Number of qubits: $O(\log(N) + b + \log(1/\gamma))$,*

5. Other elementary gates: $O(b \cdot (Q_1 + Q_2))$.

Proof. The algorithm executes the following steps. Initially, it sets $j = 2$, and $\Delta = 1/2$. At each iteration $j = 2, \dots$, the goal is to search for all gaps that are larger than $\Delta^j = \frac{1}{2^j}$.

In the first step of each iteration, a uniform grid with points $h_i = -\frac{1}{2} + i\frac{\Delta^j}{2}$, $i = 0, \dots, \frac{2}{\Delta^j}$, such that $|h_i - h_{i+1}| = \Delta^j/2$. If there exists *any* gap with width at least Δ^j , then there exists at least one point h_i such that $\sigma_{\min}(\mathbf{H} - h_i) \geq \Delta^j/4$. For each h_i in the grid, the algorithm proceeds as follows:

1. Use Lemma A.1 to create an $(a + |h_i|, b + 1, \epsilon_{\text{enc}})$ -QUBE $\mathbf{U}_i$ of $\mathbf{H} - h_i$, where we set $\epsilon_{\text{enc}} := (\frac{\Delta^j}{8})^2 \frac{1}{4a}$. The cost of $\mathbf{U}_i$ is $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}}) + O(1)$.
2. Execute QSMIN with parameters $\epsilon_{\sigma} = \frac{\Delta^{2j}}{64}$ and $\delta_{\sigma} = \frac{\delta_j}{2}$ to test if $\sigma_{\min}(\mathbf{H} - h_i)$ satisfies the desired bound.

The requirement $\epsilon_{\text{enc}} \leq \frac{\epsilon_{\sigma}}{4a}$ of Lemma 2.2 is satisfied, and therefore QSMIN returns a value $\tilde{\sigma}_i^2$ which satisfies

$$\Pr \left[\left| \tilde{\sigma}_i^2 - \sigma_{\min}^2(\mathbf{H} - h_i) \right| > \frac{\Delta^{2j}}{64} \right] \leq \frac{\delta_j}{2}.$$

This means that, for each h_i , the algorithm can test if the computed $\tilde{\sigma}_i^2$ satisfies $\tilde{\sigma}_i^2 \geq \Delta^{2j}/32$, and, if this holds for any i , then it is ensured that $\sigma_{\min}^2(\mathbf{H} - h_i) \geq \tilde{\sigma}_i^2 - \epsilon_{\sigma} > \frac{\Delta^{2j}}{64}$, or, in other words, $\sigma_{\min}(\mathbf{H} - h_i) > \frac{\Delta^j}{8}$.

Now, this lower bound allows us to count the eigenvalues smaller than h_i efficiently using QCOUNT:

3. Use Lemma A.1 to construct an $(a + |h_i|, b + 1, \epsilon'_{\text{enc}})$ -QUBE $\mathbf{U}'_i$ for $\mathbf{H} - h_i$ with $\epsilon'_{\text{enc}} := \frac{\Delta^{2j}}{64 \cdot 32 C_{\text{sgn}} a^2 N}$. Here C_{sgn} is the constant from Lemma 2.3 (note that this value of ϵ'_{enc} satisfies the requirement of the lemma). The cost of $\mathbf{U}'_i$ is $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon'_{\text{enc}}) + O(1)$.
4. By executing QCOUNT($\mathbf{U}'_i, \frac{\Delta^j}{8}, \frac{\delta_j}{2}$), we obtain z_i , the true number of eigenvalues that are smaller than h_i , with probability at least $1 - \delta_j/2$.

We can now calculate the total costs for each iteration. By replacing $\epsilon_{\sigma} = \frac{\Delta^{2j}}{64}$ and $\delta_{\sigma} = \delta_j/2 = \frac{\delta \Delta^j}{2^{j+2}}$ in Lemma 2.2, each call to QSMIN costs:

1. Queries to $\mathbf{U}_i$:

$$O \left(\frac{a^2}{\gamma \Delta^{2j}} \log \left(\frac{1}{\gamma} \right) \log \left(\frac{a}{\Delta^{2j}} \right) \log \left(\frac{\log(a/\Delta^{2j})}{1 - 2\delta_{\text{osp}}} \right) \frac{\log(2^{j+2}/(\Delta^j \delta))}{1 - 2\delta_{\text{osp}}} \right).$$

2. Queries to random state-preparation circuits $\mathbf{U}_S$:

$$O \left(\frac{1}{\gamma} \log \left(\frac{a}{\Delta^{2j}} \right) \log \left(\frac{\log(a/\Delta^{2j})}{1 - 2\delta_{\text{osp}}} \right) \frac{\log(2^{j+2}/(\Delta^j \delta))}{1 - 2\delta_{\text{osp}}} \right).$$

3. Number of qubits:

$$O \left(\log(N) + b + \log \left(\frac{1}{\gamma} \right) \right).$$

4. Other elementary gates:

$$O\left(\frac{a^2 b}{\gamma \Delta^{2j}} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\Delta^{2j}}\right) \log\left(\frac{\log(a/\Delta^{2j})}{1-2\delta_{\text{osp}}}\right) \frac{\log(2^{j+2}/(\Delta^j \delta))}{1-2\delta_{\text{osp}}}\right).$$

From Lemma 2.3, each call to $\text{QCOUNT}\left(\mathbf{U}'_i, \frac{\Delta^j}{8}, \frac{\delta \Delta^j}{2^{j+1}}\right)$ costs:

1. Queries to $\mathbf{U}'_i$: $O\left(\frac{a^2 N}{\Delta^j} \log(N) \log\left(\frac{2^j}{\Delta^j \delta}\right)\right)$,
2. Number of qubits: $O(\log(N) + b)$,
3. Other elementary gates: $O\left(b \frac{a^2 N}{\Delta^j} \log(N) \log\left(\frac{2^j}{\Delta^j \delta}\right)\right)$,
4. Total cost: $O\left((\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon'_{\text{enc}}) + b) \cdot \frac{a^2 N}{\Delta^j} \log(N) \log\left(\frac{2^j}{\Delta^j \delta}\right)\right)$.

The number of iterations can be bounded as follows. First, note that a sufficient termination criterion is $\Delta^j \leq \Delta_k$. If this holds, there exists at least one point h_i in the grid that satisfies $\sigma_{\min}(\mathbf{H} - h_i) \geq \frac{\Delta_k}{4} > \frac{\Delta^j}{4}$. Subsequently, the approximate singular value returned by QSMIN will satisfy $\tilde{\sigma}_i^2 \geq \sigma_i^2 - \Delta^{2j}/64 > \frac{1.5\Delta^{2j}}{32}$, which means that the algorithm will certainly execute QCOUNT for that point h_i . Ultimately, QCOUNT will verify that h_i is inside the correct gap, and the algorithm will terminate.

We can obtain a necessary termination criterion $\frac{\Delta^j}{4} \leq \Delta_k$ in a similar way. To see this, assume that the algorithm terminated after finding a point that satisfies $\tilde{\sigma}_i^2 > \frac{\Delta^{2j}}{32}$. This implies that $\sigma_i^2 > \frac{\Delta^{2j}}{64} \implies \sigma_i > \frac{\Delta^j}{8}$. But since h_i is inside the correct k -th gap, it means that $\Delta_k \geq 2\sigma_i \geq \frac{\Delta^j}{4}$, as initially claimed.

Since we are halving Δ^j in every iteration, we conclude that the algorithm terminates after $m := \lceil \log(1/\Delta_k) \rceil + C_{\text{gap}}$ iterations, where $C_{\text{gap}} \in [0, 2]$. This means that $\Delta^m = (1/2)^m \in \Theta(\Delta_k)$. The most expensive iteration is the last one, where the corresponding block-encodings have costs:

- $Q := \mathcal{C}(\mathbf{U}_i; \epsilon_{\text{enc}}) \in \mathcal{C}(\mathbf{U}_{\mathbf{H}}, \epsilon_{\text{enc}}) + O(1)$, where

$$\epsilon_{\text{enc}} = \left(\frac{\Delta^{\lceil \log(1/\Delta_k) \rceil + C_{\text{gap}}}}{8}\right)^2 \frac{1}{4a} \in \Theta\left(\frac{\Delta_k^2}{a}\right),$$

- $Q' := \mathcal{C}(\mathbf{U}'_i; \epsilon'_{\text{enc}}) \in \mathcal{C}(\mathbf{U}'_{\mathbf{H}}, \epsilon'_{\text{enc}}) + O(1)$, where

$$\epsilon'_{\text{enc}} = \frac{\Delta^{2\lceil \log(1/\Delta_k) \rceil + 2C_{\text{gap}}}}{64 \cdot 32a^2 C_{\text{sgn}} N} \in \Theta\left(\frac{\Delta_k^2}{a^2 N}\right).$$

The total cost of the last iteration is therefore:

$$O\left(\left[(Q + b) \cdot \frac{a^2}{\gamma \Delta^{2m}} \log\left(\frac{1}{\gamma}\right) + \mathcal{C}(\mathbf{U}_S) \frac{1}{\gamma}\right] \log\left(\frac{a}{\Delta^{2m}}\right) \log\left(\frac{\log(a/\Delta^{2m})}{1-2\delta_{\text{osp}}}\right) \frac{\log(2^{m+2}/(\Delta^m \delta))}{1-2\delta_{\text{osp}}}\right. \\ \left. + (Q' + b) \cdot \frac{a^2 N}{\Delta^m/2} \log(N) \log\left(\frac{2^m}{\Delta^m \delta}\right)\right)$$

Let us bound each additive term separately. By replacing $\Delta = 1/2$ we have:

$$\begin{aligned} O \left(\left[(Q + b) \cdot \frac{a^2}{\gamma \Delta^{2m}} \log\left(\frac{1}{\gamma}\right) + \mathcal{C}(\mathbf{U}_S) \frac{1}{\gamma} \right] \log\left(\frac{a}{\Delta^{2m}}\right) \log\left(\frac{\log(a/\Delta^{2m})}{1-2\delta_{\text{osp}}}\right) \frac{\log(2^{m+2}/(\Delta^m \delta))}{1-2\delta_{\text{osp}}} \right) \\ = O \left(\left[(Q + b) \cdot \frac{a^2}{\gamma \Delta_k^2} \log\left(\frac{1}{\gamma}\right) + \mathcal{C}(\mathbf{U}_S) \frac{1}{\gamma} \right] \log\left(\frac{a}{\Delta_k^2}\right) \log\left(\frac{\log(a/\Delta_k^2)}{1-2\delta_{\text{osp}}}\right) \frac{\log(4/\delta)}{1-2\delta_{\text{osp}}} \right). \end{aligned} \quad (5)$$

For the second term we have

$$O \left((Q' + b) \cdot \frac{a^2 N}{\Delta^m} \log(N) \log\left(\frac{2^m}{\Delta^m \delta}\right) \right) = O \left((Q' + b) \cdot \frac{a^2 N}{\Delta_k} \log(N) \log\left(\frac{1}{\Delta_k^2 \delta}\right) \right). \quad (6)$$

After simplifying some constants, the final total cost is bounded by the sum of the following terms:

- $O \left((\mathcal{C}(\mathbf{U}_i; \epsilon_{\text{enc}}) + b) \frac{a^2}{\gamma \Delta_k^2} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\Delta_k}\right) \log\left(\frac{\log(a/\Delta_k)}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/\delta)}{1-2\delta_{\text{osp}}} \log\left(\frac{1}{\Delta_k}\right) \right),$
- $O \left(\mathcal{C}(\mathbf{U}_S) \frac{1}{\gamma} \log\left(\frac{a}{\Delta_k}\right) \log\left(\frac{\log(a/\Delta_k)}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/\delta)}{1-2\delta_{\text{osp}}} \log\left(\frac{1}{\Delta_k}\right) \right),$
- $O \left((\mathcal{C}(\mathbf{U}'_i; \epsilon'_{\text{enc}}) + b) \frac{a^2 N}{\Delta_k} \log(N) \log\left(\frac{1}{\Delta_k \delta}\right) \log\left(\frac{1}{\Delta_k}\right) \right),$

where $\mathcal{C}(\mathbf{U}_i; \epsilon_{\text{enc}})$ and $\mathcal{C}(\mathbf{U}'_i; \epsilon'_{\text{enc}})$ depend on the values of ϵ_{enc} and ϵ'_{enc} , respectively, in the final iteration, and the trailing $\log\left(\frac{1}{\Delta_k}\right)$ factor is the number of iterations.

Next, we argue about the correctness of the algorithm and the approximation guarantees of the final results. Once we find some h_i , for which the result of QCOUNT satisfies $z_i = k$, we know that the point h_i is inside the correct gap Δ_k . The algorithm then returns $\tilde{\mu}_k \leftarrow h_i$ and $\tilde{\Delta}_k \leftarrow \Delta^j$, which satisfy the following bounds. In the above, we already argued that $\Delta_k \geq \Delta^j/4$ is necessary for the algorithm to terminate. But we can also see that $\Delta^j \geq \Delta_k/2$ holds as well. Indeed, the algorithm cannot terminate when $\Delta^j < \Delta_k/2$, otherwise it would have terminated in the previous step. The combination of these two facts implies that the returned Δ^m satisfies:

$$\Delta_k/2 \leq \Delta^m \leq 4\Delta_k. \quad (7)$$

Combining this with the termination criterion of the algorithm, we also conclude that the returned midpoint $\tilde{\mu}_k$ satisfies:

$$\begin{aligned} \tilde{\mu}_k &\in [\lambda_{k+1} + \Delta^m/8, \lambda_k - \Delta^m/8] \\ &\in [\lambda_{k+1} + \Delta_k/16, \lambda_k - \Delta_k/16] \\ &\in [\mu_k - 7\Delta_k/16, \mu_k + 7\Delta_k/16], \end{aligned} \quad (8)$$

where in the last we replaced $\lambda_{k+1} = \mu_k - \frac{\Delta_k}{2}$ and $\lambda_k = \mu_k + \frac{\Delta_k}{2}$.

Finally, we bound the success probability. There are a total of at most $l_j := 1 + 2/\Delta^j$ calls to QCOUNT and QSMIN in each iteration j . Each call fails with probability at most $p_j := \delta_j/2$. Thus, the total success probability for iteration j is at least

$$1 - 2 \cdot l_j p_j = 1 - 2 \cdot \frac{\delta_j}{2} \left(1 + \frac{2}{\Delta^j}\right) = 1 - \frac{\delta \Delta^j}{2^{j+1}} \left(1 + \frac{2}{\Delta^j}\right) \geq 1 - \frac{\delta}{2^j}.$$

Taking a union bound over all iterations, we have

$$p_{\text{success}} \geq 1 - \sum_{j=1}^m (\delta/2^j) \geq 1 - \delta \sum_{j=1}^{\infty} 2^{-j} = 1 - \delta.$$

□

Algorithm: $\hat{\mu}_k, \hat{\Delta}_k \leftarrow \text{QGAPconst}(\mathbf{H}, k, \delta)$

Input: Hermitian matrix $\mathbf{H} \in \mathbb{C}^{N \times N}$ with $\|\mathbf{H}\| \leq 1/2$, integer $k \in [N - 1]$, and failure probability parameter $\delta \in (0, 1/2)$. Optionally, a threshold $\Delta_{\min}$ which is used to terminate the algorithm early.

Output: Approximate gap $\tilde{\Delta}_k$ and midpoint $\tilde{\mu}_k$, satisfying the guarantees of Lemma 3.1 with probability $1 - \delta$.

```

1: Set  $j = 2, \Delta \leftarrow 1/2$ .
2: while  $\Delta^j > \Delta_{\min}$  :
3:   Construct a grid in  $[-\frac{1}{2}, \frac{1}{2}]$ , with points  $h_i = -\frac{1}{2} + i\frac{\Delta^j}{2}, i = 0, \dots, \frac{2}{\Delta^j}$ 
4:    $\delta_j \leftarrow \frac{\delta \Delta^j}{2^{j+1}}$  ▷ Failure probability for iteration  $j$ 
5:    $\epsilon_{\text{enc}} \leftarrow (\frac{\Delta^j}{8})^2 \frac{1}{4a}$ 
6:    $\epsilon'_{\text{enc}} \leftarrow \epsilon_{\text{enc}} \frac{1}{8C_{\text{sgn}} a N} = \frac{\Delta^{2j}}{64 \cdot 32 a^2 C_{\text{sgn}} N}$ 
7:    $\mathbf{U}_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon_{\text{enc}})$ 
8:    $\mathbf{U}'_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon'_{\text{enc}})$ 
9:   for all  $h_i, i = 0, \dots, \frac{2}{\Delta^j}$  :
10:     $\mathbf{U}_i \leftarrow \text{QSHIFT}(\mathbf{U}_{\mathbf{H}}, h_i, a)$ 
11:     $\tilde{\sigma}_i^2 \leftarrow \text{QSMIN}(\mathbf{U}_i, \frac{\Delta^{2j}}{64}, \frac{\delta_j}{2})$ 
12:    if  $\tilde{\sigma}_i^2 \geq \frac{\Delta^{2j}}{32}$  :
13:       $\mathbf{U}'_i \leftarrow \text{QSHIFT}(\mathbf{U}'_{\mathbf{H}}, h_i, a)$ 
14:       $z_i \leftarrow \text{QCOUNT}(\mathbf{U}'_i, \frac{\Delta^j}{8}, \frac{\delta_j}{2})$ 
15:    if  $z_i = N - k$  :
16:      return  $\tilde{\Delta}_k \leftarrow \Delta^j$ , and  $\tilde{\mu}_k \leftarrow h_i$ 
17:    $j \leftarrow j + 1$ 
18: return NULL.

```

Algorithm 1: Constant factor approximation to the k -th spectral gap and midpoint.

With some additional steps the solution accuracy can be increased to any $\epsilon > 0$.

Theorem 3.1. *Let $\mathbf{H}$ be an $N \times N$ Hermitian matrix, $\|\mathbf{H}\| \leq 1/2$, $k \in [N - 1]$ be an integer, and $\epsilon, \delta \in (0, 1)$ an accuracy and failure probability parameter, respectively. Let*

- $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$ denote the cost of the unitary $\mathbf{U}_{\mathbf{H}}$ which forms an $(a, b, \epsilon_{\text{enc}})$ -QUBE of $\mathbf{H}$.
- Π be a $(\gamma, \delta_{\text{osp}})$ -OSP distribution.

Algorithm 2 returns $\tilde{\lambda}_k, \tilde{\lambda}_{k+1}, \tilde{\mu}_k, \tilde{\Delta}_k$ that satisfy:

$$\begin{aligned} \tilde{\lambda}_k &\in [\lambda_k - \epsilon \Delta_k / 2, \lambda_k], & \tilde{\lambda}_{k+1} &\in [\lambda_{k+1}, \lambda_{k+1} + \epsilon \Delta_k / 2] \\ \tilde{\mu}_k &\in \mu_k \pm \epsilon \Delta_k / 2, & \tilde{\Delta}_k &\in (1 \pm \epsilon) \Delta_k. \end{aligned}$$

In addition to the costs induced by calling Algorithm 1 in the first step, it requires

1. Queries to unitaries with complexity $O(\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}}))$, where $\epsilon_{\text{enc}} \in \Theta(\Delta_k^2 / a)$:

$$Q_1 := O\left(\frac{a^2}{\gamma \epsilon^2 \Delta_k^2} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon \Delta_k}\right) \log\left(\frac{\log(a/(\epsilon \Delta_k))}{1 - 2\delta_{\text{osp}}}\right) \frac{\log(1/(\epsilon \delta))}{1 - 2\delta_{\text{osp}}}\right),$$

2. Queries to OSP circuits $\mathbf{U}_S \sim \mathbf{\Pi}$:

$$O\left(\frac{1}{\gamma} \log\left(\frac{a}{\epsilon \Delta_k}\right) \log\left(\frac{\log(a/(\epsilon \Delta_k))}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/(\epsilon \delta))}{1-2\delta_{\text{osp}}}\right),$$

3. Queries to unitaries with complexity $Q_2 := O(\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon'_{\text{enc}}))$, where $\epsilon'_{\text{enc}} \in \Theta(\epsilon^2 \Delta_k^2 / (a^2 N))$:

$$O\left(\frac{a^2 N}{\epsilon^2 \Delta_k} \log(N) \log\left(\frac{1}{\delta}\right)\right),$$

4. Number of qubits: $O\left(\log(N) + b + \log\left(\frac{1}{\gamma}\right)\right)$,

5. Other elementary gates: $O(b \cdot (Q_1 + Q_2))$.

If we use Lemma 2.1 for $\mathbf{\Pi}$, after some simplifications, the total circuit complexity (of both Alg. 1 and 2) is upper bounded by

$$O\left(\left[N^2 + N \cdot \frac{a^2 b \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_1)}{\epsilon^2 \Delta_k^2}\right] \text{polylog}\left(N, a, \epsilon^{-1}, \Delta_k^{-1}, \delta^{-1}\right)\right), \quad (9)$$

where $\epsilon_1 \in \Theta\left(\frac{\epsilon^2 \Delta_k^2}{a^2 N}\right)$.

Proof. Let us call $\hat{\mu}_k$ and $\hat{\Delta}_k$ the initial solutions returned by Algorithm 1, i.e., the solutions that satisfy the constant factor accuracy of Lemma 3.1. We know that $\hat{\Delta}_k \in [0.5 \Delta_k, 4 \Delta_k]$. By setting $\epsilon_k := \epsilon \hat{\Delta}_k / 4$, it means that $\epsilon_k \leq \epsilon \Delta_k$.

We now approximate both eigenvalues λ_k and λ_{k+1} to accuracy $\epsilon_k/2$ each as follows. Consider first λ_{k+1} . We make a grid on the left of $\hat{\mu}_k$ with step $\epsilon_k/4$, consisting of the points $l_i = \hat{\mu}_k + i\epsilon_k/4$, for $i = -\lceil 2\hat{\Delta}_k/\epsilon_k \rceil, \dots, -2, -1$. Note that, there exists at least one point $l^* > \lambda_{k+1}$ such that $\epsilon_k/4 \leq |\lambda_{k+1} - l^*| < \epsilon_k/2$, or, equivalently, $\sigma_{\min}(\mathbf{H} - l^*) \in [\epsilon_k/4, \epsilon_k/2]$.

To find such a point, we first set the parameters:

$$\epsilon_\sigma \leftarrow \frac{3\epsilon_k^2}{64}, \quad \epsilon_{\text{enc}} \leftarrow \frac{\epsilon_\sigma}{4a} = \frac{3\epsilon_k^2}{256a}, \quad \delta_\sigma \leftarrow \frac{\delta}{\lceil 2\hat{\Delta}_k/\epsilon_k \rceil}, \quad \epsilon'_{\text{enc}} \leftarrow \frac{\epsilon_k^2}{64 \cdot 32a^2 C_{\text{sgn}} N},$$

and prepare two block-encodings:

- $\mathbf{U}_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon_{\text{enc}})$,
- $\mathbf{U}'_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon'_{\text{enc}})$.

Then we execute the steps 10-18 for each i , similar to Algorithm 1, but with the aforementioned parameters. We can verify that the returned values $\tilde{\lambda}_k, \tilde{\lambda}_{k+1}, \tilde{\mu}_k, \tilde{\Delta}_k$, satisfy the advertised bounds.

It remains to bound the complexity. There are a total of $O(\epsilon^{-1})$ points in the two grids, which gives the total number of calls to the subroutines QSMIN and QCOUNT. At each grid point, we call QENC with parameter $\epsilon_{\text{enc}} = 3\epsilon_k^2/256a \in \Theta(\Delta_k^2/a)$, followed by QSMIN with parameters:

$$\epsilon_\sigma = \frac{3\epsilon_k^2}{64} \in \Theta(\epsilon^2 \Delta_k^2), \quad \text{and} \quad \delta_\sigma = \frac{\delta}{\lceil 2\hat{\Delta}_k/\epsilon_k \rceil} \in \Theta(\epsilon \delta).$$

The cost for each QSMIN call is

$$1. \ O\left(\frac{a^2}{\gamma \epsilon^2 \Delta_k^2} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon \Delta_k}\right) \log\left(\frac{\log(a/(\epsilon \Delta_k))}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/(\epsilon \delta))}{1-2\delta_{\text{osp}}}\right) \text{ queries to } \mathbf{U}_{\mathbf{H}},$$

2. $O\left(\frac{1}{\gamma} \log\left(\frac{a}{\epsilon \Delta_k}\right) \log\left(\frac{\log(a/(\epsilon \Delta_k))}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/(\epsilon \delta))}{1-2\delta_{\text{osp}}}\right)$ queries to OSP circuits $\mathbf{U}_S \sim \mathbf{\Pi}$,
3. $O\left(\log(N) + b + \log\left(\frac{1}{\gamma}\right)\right)$ qubits,
4. $O\left(\frac{a^2}{\gamma \epsilon^2 \Delta_k^2} \log\left(\frac{1}{\gamma}\right) \log\left(\frac{a}{\epsilon \Delta_k}\right) \log\left(\frac{\log(a/(\epsilon \Delta_k))}{1-2\delta_{\text{osp}}}\right) \frac{\log(1/(\epsilon \delta))}{1-2\delta_{\text{osp}}}\right)$ other elementary gates.

In the worst case, QCOUNT will also be called in every iteration. Each call takes as input an $(a, b, \epsilon'_{\text{enc}})$ -QUBE, accuracy $\epsilon'_{\text{enc}} = \epsilon_k^2 / (64a^2 C_{\text{sgn}} N) \in \Theta(\epsilon^2 \Delta_k^2 / (a^2 N))$, and failure probability $\delta_\sigma = \Theta(\epsilon \delta)$. Substituting in Lemma 2.3, each QCOUNT call costs:

$$O\left(\frac{a^2 N}{\epsilon^2 \Delta_k} \log(N) \log\left(\frac{1}{\delta}\right)\right)$$

calls to the block-encoding circuit $\mathbf{U}'_{\mathbf{H}}$. □

Algorithm: $[\tilde{\lambda}_k, \tilde{\lambda}_{k+1}, \tilde{\mu}_k, \tilde{\Delta}_k] \leftarrow \text{QEIG}(\mathbf{H}, k, \delta, \epsilon)$

Input: Hermitian matrix $\mathbf{H} \in \mathbb{C}^{N \times N}$, $\|\mathbf{H}\| \leq 1/2$, integer $k \in [N-1]$, failure probability parameter $\delta \in (0, 1/2)$, accuracy parameter $\epsilon \in (0, 1)$.

Output: Approximate gap $\tilde{\Delta}_k$, midpoint $\tilde{\mu}_k$, and eigenvalues $\tilde{\lambda}_k, \tilde{\lambda}_{k+1}$, satisfying the guarantees of Theorem 3.1 with probability at least $1 - \delta$.

- 1: $\hat{\mu}_k, \hat{\Delta}_k \leftarrow \text{QGAPconst}(\mathbf{H}, k, \delta/2)$
- 2: $\epsilon_k \leftarrow \epsilon \hat{\Delta}_k / 4$
- 3: $\epsilon_\sigma \leftarrow \frac{3\epsilon_k^2}{64}$
- 4: $\delta_\sigma \leftarrow \frac{\delta}{\lceil 2\hat{\Delta}_k / \epsilon_k \rceil}$ ▷ Failure probability per iteration
- 5: $\epsilon_{\text{enc}} \leftarrow \frac{\epsilon_\sigma}{4a} = \frac{3\epsilon_k^2}{256a}$
- 6: $\epsilon'_{\text{enc}} \leftarrow \frac{\epsilon_k^2}{64 \cdot 32a^2 CN}$
- 7: $\mathbf{U}_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon_{\text{enc}})$
- 8: $\mathbf{U}'_{\mathbf{H}} \leftarrow \text{QENC}_{a,b}(\mathbf{H}, \epsilon'_{\text{enc}})$
- 9: $M \leftarrow \lceil 2\hat{\Delta}_k / \epsilon_k \rceil$
- 10: **for** $i = -M, -M+1, \dots, M-1, M$:
- 11: $l_i \leftarrow \hat{\mu}_k + i\epsilon_k/4$
- 12: $\mathbf{U}_i \leftarrow \text{QSHIFT}(\mathbf{U}_{\mathbf{H}}, l_i, a)$
- 13: $\tilde{\sigma}_i^2 \leftarrow \text{QSMIN}(\mathbf{U}_i, \epsilon_\sigma, \delta_\sigma)$
- 14: **if** $\tilde{\sigma}_i^2 \geq \frac{\epsilon_k^2}{32}$:
- 15: $\mathbf{U}'_i \leftarrow \text{QSHIFT}(\mathbf{U}'_{\mathbf{H}}, l_i, a)$
- 16: $z_i \leftarrow \text{QCOUNT}(\mathbf{U}'_i, \frac{\epsilon_k^2}{64}, \delta_\sigma)$
- 17: **else:**
- 18: $z_i \leftarrow -1$
- 19: $\tilde{\lambda}_k \leftarrow \min_i \{l_i \mid z_i = N - k\}$, and $\tilde{\lambda}_{k+1} \leftarrow \max_i \{l_i \mid z_i = N - k\}$
- 20: $\tilde{\mu}_k \leftarrow (\tilde{\lambda}_k + \tilde{\lambda}_{k+1})/2$
- 21: $\tilde{\Delta}_k \leftarrow \tilde{\lambda}_k - \tilde{\lambda}_{k+1}$
- 22: **return** $\tilde{\lambda}_k, \tilde{\lambda}_{k+1}, \tilde{\mu}_k, \tilde{\Delta}_k$

Algorithm 2: Main algorithm for high-accuracy consecutive eigenvalue pair, spectral gap, and midpoint.

4 Complexity in QRAM

In this section we specialize the complexity analysis in the QRAM model [58]. We describe a specific technique to construct block-encodings via the so-called *quantum-accessible* data structures, studied in [41, 49, 50, 65]. We assume a dense $N \times N$ matrix $\mathbf{A} \in \mathbb{C}$, whose elements are known (i.e. pre-computed), and stored as an array in a classical memory. Given this classical array, a data structure of the form of [41] can be constructed in an *offline* fashion and stored in a classical memory in $O(N^2)$ arithmetic operations, and it requires $O(N^2 \log(N))$ space, i.e. words, assuming each matrix element requires one word to store.

For dense matrices, the data structure consists of N binary trees with $O(\log(N))$ depth and N leaves each. If we were to prepare a quantum circuit for the access oracle of this data structure, we would need $O(N^2)$ gates to implement it. To elaborate more, each binary tree B_i can be obtained by constructing a circuit that rotates the state $|0\rangle^{\otimes \log(N)}$ to the state $|B_i\rangle$, where the latter contains the N leaves of the tree B_i . Such a circuit can be constructed using $O(N)$ one and two qubit gates (rotations and CNOT gates) [66, 67], and we need such a circuit for every $i \in [N]$. However, if the quantum computer has the ability to query the elements of the classical data structure in superposition, which is possible in the QRAM model, the gate complexity can be reduced to $O(\text{polylog}(N))$ [41]. In particular, Theorem 10 of [41] showed that this construction enables efficient circuits implementing the maps

$$|i\rangle |0\rangle \rightarrow |i\rangle |\mathbf{A}_i\rangle \quad \text{and} \quad |0\rangle |j\rangle \rightarrow |\tilde{\mathbf{A}}\rangle |j\rangle, \quad (10)$$

where $|\mathbf{A}_i\rangle$ is the normalized quantum state corresponding to the i -th row of $\mathbf{A}$ and $\langle i|\tilde{\mathbf{A}} = \|\mathbf{A}_i\|$ encodes the row norms. These circuits can be implemented to precision ϵ with gate complexity $O(\text{polylog}(N/\epsilon))$ [41, 50]. The following lemma can be used to construct a block-encoding from a quantum-accessible data structure in QRAM.

Lemma 4.1 (Imported result from [50, 49]). *Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ be a matrix stored in a quantum-accessible data structure. There exists a $\text{QENC}_{a,b}(\mathbf{A}, \epsilon)$ subroutine with $a = \|\mathbf{A}\|_F$, $b = \lceil \log(N) \rceil + 1$, and cost $\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \epsilon) = O(\text{polylog}(N/\epsilon))$.*

By using the specific block-encodings from Appendix A and state preparation circuits from Section 2, we can bound the complexity of the algorithm as follows.

Corollary 4.1. *Using Lemma 4.1 to construct block-encodings, and Lemma 2.1 for state preparation, the total complexity of Theorem 3.1 becomes:*

$$O\left(\frac{N^2 b}{\epsilon^2 \Delta_k^2} \text{polylog}(N, \delta^{-1}, \epsilon^{-1}, \Delta_k^{-1})\right),$$

and the algorithm succeeds with probability at least $1 - \delta$.

Proof. It suffices to replace the costs of the block-encodings in Theorem 3.1. For all block-encodings we can use Lemma 4.1, which means that $a = \|\mathbf{H}\|_F \in O(\sqrt{N})$. The cost has a polylogarithmic dependence on the accuracy. Replacing the costs in Eq. (9) we have

$$\begin{aligned} \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_1) &\in \text{polylog}(N/(\epsilon \Delta_k)), \\ \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_2) &\in \text{polylog}(N/\Delta_k). \end{aligned}$$

Combining everything together, Eq. (9) becomes

$$O\left(N^2 b \left(\frac{1}{\epsilon^2 \Delta_k^2} + \frac{1}{\Delta_k^2}\right) \text{polylog}\left(N, a, \epsilon^{-1}, \Delta_k^{-1}, \delta^{-1}\right)\right),$$

where we absorbed all polylogarithmic factors in the trailing term. $\square$

5 A lower bound in the Black-Box model

In this section we derive a lower bound for the problem of deciding the existence of a spectral gap of a binary matrix, in the so-called Black-Box access model. Here we assume a unitary $\mathbf{O}_{\mathbf{A}}$ through which we can query elements of the matrix $\mathbf{A}$ via:

$$\mathbf{O}_{\mathbf{A}} : |i\rangle |j\rangle |0\rangle \mapsto |i\rangle |j\rangle |\mathbf{A}_{i,j}\rangle.$$

Several lower bounds are known for related problems. A straightforward query lower bound for determining the midpoint μ_k can be obtained from the median problem [68]. Let $\mathcal{A}$ be a quantum algorithm which, given $\mathbf{O}_{\mathbf{A}}$, and an integer k , it returns any (classical) value $\tilde{\mu}_k$ such that $\tilde{\mu}_k \in (\lambda_k, \lambda_{k+1})$. Then $\mathcal{A}$ can be directly used to solve the median problem, where, given oracle access to a sequence of N numbers $\{x_1, \dots, x_N\}$, where N is even, we seek to find a number that lies strictly inside $x \in (x_{N/2}, x_{N/2} + 1)$. Such a sequence can be seen as the elements of a diagonal matrix, and the eigenvalues of such a matrix are the diagonal elements themselves. [68] proved a $\Omega(N)$ query lower bound for the median problem, assuming that we have oracle access to a quantum circuit implementing the sequence. Indeed, the same bound holds even if we only want to verify that $\tilde{\mu}_k$ is inside the correct interval, since this generalizes the counting problem [68, 69, 70].

Here we obtain a stronger lower bound for deciding whether there exists a gap between two consecutive eigenvalues, inspired by the reductions of [56, 57]. We remark that the construction below is inherently non-Hermitian. Whether analogous black-box lower bounds hold for Hermitian matrices remains an open question.

Theorem 5.1. *Given a matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$, a unitary $\mathbf{O}_{\mathbf{A}} : |i\rangle |j\rangle |0\rangle \mapsto |i\rangle |j\rangle |\mathbf{A}_{i,j}\rangle$, and an integer k , any quantum algorithm which answers *TRUE* if $\Delta_k := |\lambda_k(\mathbf{A}) - \lambda_{k+1}(\mathbf{A})| > 0$, and *FALSE* otherwise, must execute $\Omega(N^2)$ queries to $\mathbf{O}_{\mathbf{A}}$. Here the eigenvalues of $\mathbf{A}$ are order such that $\text{Re}(\lambda_1) \geq \text{Re}(\lambda_2) \geq \dots \geq \text{Re}(\lambda_N)$, and if $\text{Re}(\lambda_k) = \text{Re}(\lambda_{k+1})$, then $\text{Im}(\lambda_k) \geq \text{Im}(\lambda_{k+1})$.*

Proof. The proof closely follows the techniques of [56, 57] for other similar problems (rank, determinant, matrix power, and inverse), essentially by reducing the problem to counting the number of triangles in a directed graph. Assume that we are given a set $X \in \{0, 1\}^{N^2}$ and the value $a = \lceil N^2/2 \rceil$, and we ask whether $\sum_{i=1}^{N^2} X_i = a$. The query complexity of this problem is $\Omega(N^2)$ [56].

For the reduction the first step is to create a couple of graphs. We create a bipartite graph G , with N sources s_i and N targets t_j , and we insert an edge between s_i and t_j iff $X_{iN+j} = 1$. We also create a utility node v , which has N outgoing edges to every one of the sources, and N incoming edges from every one of the targets.

Thereafter, we construct a second bipartite graph G' which also has N sources s'_i , N targets t'_j , and a utility node v' . We create exactly a edges between the sources and the targets, in a round-robin manner, so that the edges are “evenly distributed” among the two vertex sets (for each s_i , we add roughly $\lceil a/N \rceil$ outgoing edges). Finally, we add N edges from v' to every s'_i , and N edges from every t'_j to v' .

Note that the two graphs are disjoint. Consider the larger graph $F = (V \cup V', E \cup E')$, and its adjacency matrix $\mathbf{A}$. F has $4N + 2$ nodes. If we choose the node labels of F appropriately, $\mathbf{A}$ consists of two diagonal blocks: one block which corresponds to the adjacency of G , and the other to the one of G' . More specifically,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_G & \\ & \mathbf{A}_{G'} \end{pmatrix}.$$

We will show that $\mathbf{A}_G$ and $\mathbf{A}_{G'}$ have only three nonzero eigenvalues.

The elements of the matrix $\mathbf{A}^3$ contain the number of paths of length three starting from every node i to every other node j . Since the only length-three paths from v and v' are towards themselves, the matrix $\mathbf{A}^3$ has the following structure:

$$\mathbf{A}^3 = \begin{pmatrix} \sigma & & \\ & \mathbf{B} & \\ & & a \\ & & & \mathbf{B}' \end{pmatrix},$$

where a is located in position $\mathbf{A}_{v',v'}^3$ and $\sigma := \sum_{i=1}^{N^2} X_i$ in position $\mathbf{A}_{v,v}^3$. It is not hard to see that σ and a are eigenvalues, since $\det(\sigma \mathbf{I} - \mathbf{A}^3) = \det(a \mathbf{I} - \mathbf{A}^3) = 0$.

The matrices $\mathbf{B}$ and $\mathbf{B}'$ have a special structure. Let us analyze $\mathbf{B}$ first. From each source vertex $s_i, i \in [N]$, there are exactly $d_i = \text{outdeg}(s_i)$ paths of length three to every source s_j , including itself. This means that the row i of $\mathbf{B}$ contains exactly N non-zeros, with the same value d_i , at the positions corresponding to the sources. This gives rise to a submatrix $\mathbf{B}_s$. Looking at the targets t_l , note that there are exactly $f_l = \text{indeg}(t_l)$ paths from every node $t_m, m \in [N]$, to t_l . This means that the l -th column of the matrix $\mathbf{B}$ has exactly N nonzeros, equal to f_l , at the positions that correspond to target vertices. This gives rise to a submatrix $\mathbf{B}_t$, where $\mathbf{B}_s$ and $\mathbf{B}_t$ are disjoint. With this description, we conclude that the matrix $\mathbf{B}$ has the following structure:

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_s & \\ & \mathbf{B}_t \end{pmatrix},$$

where both $\mathbf{B}_s$ and $\mathbf{B}_t$ have size $N \times N$. It turns out that they are rank-one matrices. Indeed, we can write $\mathbf{B}_s = \mathbf{d}_s \mathbf{e}^\top$, where $\mathbf{e}$ is the all-ones vector, $\mathbf{d}_s^\top = (d_1 \ \dots \ d_N)$, and $\mathbf{B}_t = \mathbf{e} \mathbf{f}_t^\top$, where $\mathbf{f}_t^\top = (f_1 \ \dots \ f_N)$. The only nonzero eigenvalues of $\mathbf{B}_s$ and $\mathbf{B}_t$ are equal to $\lambda_s = \mathbf{d}_s^\top \mathbf{e} = \sum_{i=1}^N d_i$ and $\lambda_f = \mathbf{f}_t^\top \mathbf{e} = \sum_{i=1}^N f_i$. We can see that

$$\sigma = \lambda_s = \lambda_f.$$

Next, we shift our attention to $\mathbf{B}'$. Since it has very similar structure to $\mathbf{B}$, we can argue accordingly that the only two non-zero eigenvalues of $\mathbf{B}'$ are equal to a . Now, it follows that

$$\Lambda(\mathbf{A}^3) = \{a\} \cup \{\sigma\} \cup \Lambda(\mathbf{B}) \cup \Lambda(\mathbf{B}') = \{a, a, a, \sigma, \sigma, \sigma\}.$$

The above imply that $\mathbf{A}^3$ is diagonalizable. The eigenvalues of $\mathbf{A}$ are the third-roots of the eigenvalues of $\mathbf{A}^3$. Since $\mathbf{A}$ is real, its complex eigenvalues come in conjugate pairs. The spectrum of $\mathbf{A}$ is equal to

$$\Lambda(\mathbf{A}) = \{a_1, a_2, a_3, \sigma_1, \sigma_2, \sigma_3\},$$

where $\{a_1, a_2, a_3\}$ are either all equal to $|a|^{1/3}$ or equal to $|a|^{1/3} \times \{z_1, z_2, z_3\}$ where z_1, z_2, z_3 are the three third-roots of unity. Similarly, $\{\sigma_1, \sigma_2, \sigma_3\}$ are either all equal to $|\sigma|^{1/3}$ or $|\sigma|^{1/3} \times \{z_1, z_2, z_3\}$.

Let $\{\lambda_1, \lambda_2, \dots, \lambda_6\}$ be the eigenvalues of $\mathbf{A}$ sorted in descending order of their real parts, where conflicts are resolved with respect to the imaginary part, i.e.

$$\text{Re}(\lambda_1) \geq \text{Re}(\lambda_2) \geq \dots \geq \text{Re}(\lambda_6).$$

We now derive the final argument.

($\implies$) : If $a = \sigma$, we have the following four cases for the eigenvalues of $\mathbf{A}$:

a	σ	Relation
Real	Real	$\lambda_i = \lambda_j, j \in [6]$.
Real	Complex	$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 \neq \lambda_5 \neq \lambda_6$.
Complex	Real	Same as above.
Complex	Complex	$\lambda_1 = \lambda_2 \neq \lambda_3 = \lambda_4 \neq \lambda_5 = \lambda_6$.

($\Leftarrow$) : On the opposite direction, we prove each case separately:

- If $\lambda_i = \lambda_{i+1}$, for $i = 1, \dots, 5$, all eigenvalues are equal and hence $a = \sigma$.
- $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 \neq \lambda_5 \neq \lambda_6$. We already argued that a_i are either all equal to each other or equal to the scaled roots of unity. The same for σ_i . The only way that the top-three λ_i are equal is if either all a_i are equal and real, or all σ_i . Since they are real, and equal to λ_4 , it means that λ_4 is also real. We conclude that at least one of $\lambda_1, \dots, \lambda_4$ is equal to a , and at least one of them is equal to σ , therefore $a = \sigma$.
- The only way that top-6 eigenvalues are equal in pairs, is if both a_i and σ_i are the scaled roots of unity, and, in this case, with the same scaling factor, which also implies that $a = \sigma$.

We conclude that $a = \sigma$ if and only if one of the following relation holds:

- $\lambda_i = \lambda_{i+1}, i = 1, \dots, 5$.
- $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 \neq \lambda_5 \neq \lambda_6$.
- $\lambda_1 = \lambda_2 \neq \lambda_3 = \lambda_4 \neq \lambda_5 = \lambda_6$.

Therefore, one can call the gap algorithm five times, for $k = 1, \dots, 5$, and revisit the three cases above in order to verify whether $a = \sigma$. $\square$

6 Conclusion

In this work we studied algorithms for computing spectral gaps between eigenvalues of Hermitian matrices and the corresponding midpoints. One of the main subroutines is a randomized state preparation method, which achieves at least $1/N$ overlap with any fixed target state with probability at least $3/5$, and requires $O(N)$ classical random bits to prepare. This can be used to efficiently approximate the smallest singular value of a block-encoded matrix with high probability, and to count the eigenvalues of a Hermitian matrix that are smaller than a given threshold. In the quantum-accessible data structure model, the total query complexity of our main algorithm scales at most quadratically with respect to the problem size N . This indicates the potential for a polynomial quantum speed-up, since the complexity of the best-known classical algorithms require $\tilde{O}(N^\omega)$ arithmetic operations. In the black-box access model, we also proved an $\Omega(N^2)$ query lower bound for deciding whether there is a gap between two consecutive eigenvalues of a binary matrix.

References

- [1] Alex N. Beavers and Eugene D. Denman. “A new similarity transformation method for eigenvalues and eigenvectors”. *Mathematical Biosciences* **21**, 143–169 (1974).
- [2] Yuji Nakatsukasa and Nicholas J. Higham. “Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD”. *SIAM Journal on Scientific Computing* **35**, A1325–A1349 (2013).

- [3] Jess Banks, Jorge Garza-Vargas, Archit Kulkarni, and Nikhil Srivastava. “Pseudospectral shattering, the sign function, and diagonalization in nearly matrix multiplication time”. *Foundations of Computational Mathematics* **23**, 1959–2047 (2023).
- [4] Rikhav Shah. “Hermitian diagonalization in linear precision”. In Proc. ACM-SIAM Symposium on Discrete Algorithms. Pages 5599–5615. SIAM (2025).
- [5] Ian Jolliffe. “Principal Component Analysis”. Springer-Verlag. (2002).
- [6] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Kernel Principal Component Analysis”. In International Conference on Artificial Neural Networks. Pages 583–588. Springer Berlin Heidelberg (1997).
- [7] Ulrike von Luxburg. “A tutorial on spectral clustering”. *Statistics and Computing* **17**, 395–416 (2007).
- [8] Jeff Cheeger. “A lower bound for the smallest eigenvalue of the Laplacian”. In Problems in analysis: A symposium in honor of Salomon Bochner. Pages 195–199. Princeton University Press, Princeton, NJ (1970).
- [9] Noga Alon and Vitali D. Milman. “ λ_1 , Isoperimetric inequalities for graphs, and superconcentrators”. *Journal of Combinatorial Theory, Series B* **38**, 73–88 (1985).
- [10] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. “Many sparse cuts via higher eigenvalues”. In Proc. ACM Symposium on Theory of Computing. Pages 1131–1140. (2012).
- [11] Subir Sachdev. “Quantum phase transitions”. Cambridge University Press. (2011). 2nd edition.
- [12] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. “A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem”. *Science* **292**, 472–475 (2001).
- [13] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. “Self-consistent-field calculations using Chebyshev-filtered subspace iteration”. *Journal of Computational Physics* **219**, 172–184 (2006).
- [14] Georg Kresse and Jürgen Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. *Physical Review B* **54**, 11169 (1996).
- [15] Joost VandeVondele, Urban Borstnik, and Jurg Hutter. “Linear scaling self-consistent field calculations with millions of atoms in the condensed phase”. *Journal of Chemical Theory and Computation* **8**, 3565–3573 (2012).
- [16] Lin Lin, Jianfeng Lu, and Lexing Ying. “Numerical methods for Kohn–Sham density functional theory”. *Acta Numerica* **28**, 405–539 (2019).
- [17] Taehee Ko, Xiantao Li, and Chunhao Wang. “Implementation of the density-functional theory on quantum computers with linear scaling with respect to the number of atoms” (2023).
- [18] Martin Ziegler and Vasco Brattka. “Computability in linear algebra”. *Theoretical Computer Science* **326**, 187–211 (2004).
- [19] Aleksandros Sobczyk, Marko Mladenović, and Mathieu Luisier. “Invariant subspaces and PCA in nearly matrix multiplication time”. *Advances in Neural Information Processing Systems* **37**, 19013–19086 (2024).
- [20] Aleksandros Sobczyk. “Deterministic complexity analysis of hermitian eigenproblems”. In 52nd International Colloquium on Automata, Languages, and Programming. Volume 334, pages 131:1–131:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2025).
- [21] James Demmel, Ioana Dumitriu, and Ryan Schneider. “Generalized pseudospectral shattering and inverse-free matrix pencil diagonalization”. *Foundations of Computational Mathematics* **24**, 1–77 (2024).

- [22] James Demmel, Ioana Dumitriu, and Ryan Schneider. “Fast and inverse-free algorithms for deflating subspaces”. *Linear Algebra and its Applications* **735**, 222–258 (2026).
- [23] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. “More asymmetry yields faster matrix multiplication”. In Proc. ACM-SIAM Symposium on Discrete Algorithms. Pages 2005–2039. SIAM (2025).
- [24] Ming Gu and Stanley C. Eisenstat. “A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem”. *SIAM Journal on Matrix Analysis and Applications* **16**, 172–191 (1995).
- [25] Daniel A. Spielman and Shang-Hua Teng. “Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems”. *SIAM Journal on Matrix Analysis and Applications* **35**, 835–885 (2014).
- [26] Ioannis Koutis, Alex Levin, and Richard Peng. “Faster spectral sparsification and numerical algorithms for SDD matrices”. *ACM Transactions on Algorithms* **12**, 1–16 (2015).
- [27] Michele Benzi, Michele Rinelli, and Igor Simunec. “Estimation of spectral gaps for sparse symmetric matrices” (2024).
- [28] Michael F. Hutchinson. “A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines”. *Communications in Statistics - Simulation and Computation* **19**, 433–450 (1990).
- [29] Haim Avron and Sivan Toledo. “Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix”. *Journal of the ACM* **58**, 1–34 (2011).
- [30] Farbod Roosta-Khorasani and Uri Ascher. “Improved bounds on sample size for implicit matrix trace estimators”. *Foundations of Computational Mathematics* **15**, 1187–1212 (2015).
- [31] Alice Cortinovis and Daniel Kressner. “On randomized trace estimates for indefinite matrices with an application to determinants”. *Foundations of Computational Mathematics* **22**, 875–903 (2022).
- [32] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. “The Variational Quantum Eigensolver: A review of methods and best practices”. *Physics Reports* **986**, 1–128 (2022).
- [33] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O’Brien. “A variational eigenvalue solver on a photonic quantum processor”. *Nature Communications* **5**, 4213 (2014).
- [34] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. “The theory of variational hybrid quantum-classical algorithms”. *New Journal of Physics* **18**, 023023 (2016).
- [35] A. Yu. Kitaev. “Quantum measurements and the Abelian Stabilizer Problem” (1995).
- [36] Daniel S. Abrams and Seth Lloyd. “Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors”. *Physical Review Letters* **83**, 5162 (1999).
- [37] Uwe Dorner, Rafal Demkowicz-Dobrzanski, Brian J. Smith, Jeff S. Lundeen, Wojciech Wasilewski, Konrad Banaszek, and Ian A. Walmsley. “Optimal quantum phase estimation”. *Physical Review Letters* **102**, 040403 (2009).
- [38] Lin Lin and Yu Tong. “Near-optimal ground state preparation”. *Quantum* **4**, 372 (2020).
- [39] Guang Hao Low and Yuan Su. “Quantum Eigenvalue Processing”. *SIAM Journal on Computing* **55**, 135–215 (2026).

- [40] Alex Kerzner, Vlad Gheorghiu, Michele Mosca, Thomas Guilbaud, Federico Carmi-nati, Fabio Fracas, and Luca Dellantonio. “A square-root speedup for finding the smallest eigenvalue”. *Quantum Science and Technology* **9**, 045025 (2024).
- [41] Iordanis Kerenidis and Anupam Prakash. “Quantum recommendation systems”. In *Proc. Innovations in Theoretical Computer Science. Volume 67, pages 49:1–49:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik* (2017).
- [42] Changpeng Shao. “Computing eigenvalues of diagonalizable matrices on a quantum computer”. *ACM Transactions on Quantum Computing* **3**, 1–20 (2022).
- [43] Jeffrey B. Parker and Ilon Joseph. “Quantum phase estimation for a class of generalized eigenvalue problems”. *Physical Review A* **102**, 022422 (2020).
- [44] Zhiyan Ding, Haoya Li, Lin Lin, HongKang Ni, Lexing Ying, and Ruizhe Zhang. “Quantum multiple eigenvalue gaussian filtered search: an efficient and versatile quan-tum phase estimation method”. *Quantum* **8**, 1487 (2024).
- [45] Zhiyan Ding and Lin Lin. “Simultaneous estimation of multiple eigenvalues with short-depth quantum circuit on early fault-tolerant quantum computers”. *Quantum* **7**, 1136 (2023).
- [46] Simon Apers and Ronald de Wolf. “Quantum speedup for graph sparsification, cut approximation, and Laplacian solving”. *SIAM Journal on Computing* **51**, 1703–1742 (2022).
- [47] Yanlin Chen, András Gilyén, and Ronald de Wolf. “A quantum speed-up for approxi-mating the top eigenvectors of a matrix”. In *Proc. ACM-SIAM Symposium on Discrete Algorithms. Pages 994–1036. SIAM* (2025).
- [48] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. “Quantum simulation of electronic structure with linear depth and connectivity”. *Physical Review Letters* **120**, 110501 (2018).
- [49] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. “Quantum singu-lar value transformation and beyond: exponential improvements for quantum matrix arithmetics”. In *Proc. ACM Symposium on Theory of Computing. Pages 193–204. (2019).*
- [50] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. “The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation”. In *Proc. International Colloquium on Automata, Languages, and Pro-gramming. Volume 132, pages 33:1–33:14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik* (2019).
- [51] Charles Kenney and Alan J Laub. “Rational iterative methods for the matrix sign function”. *SIAM Journal on Matrix Analysis and Applications* **12**, 273–291 (1991).
- [52] Nir Ailon and Bernard Chazelle. “The fast Johnson–Lindenstrauss transform and approximate nearest neighbors”. *SIAM Journal on Computing* **39**, 302–322 (2009).
- [53] Nam H. Nguyen, Thong T. Do, and Trac D. Tran. “A fast and efficient algorithm for low-rank approximation of a matrix”. In *Proc. ACM Symposium on Theory of Computing. Pages 215–224. (2009).*
- [54] Joel A Tropp. “Improved analysis of the subsampled randomized Hadamard trans-form”. *Advances in Adaptive Data Analysis* **3**, 115–126 (2011).
- [55] Andris Ambainis. “Quantum lower bounds by quantum arguments”. In *Proc. ACM Symposium on Theory of Computing. Pages 636–643. (2000).*
- [56] Sebastian Dörn and Thomas Thierauf. “The quantum query complexity of the deter-minant”. *Information Processing Letters* **109**, 325–328 (2009).
- [57] Sander Gribling, Iordanis Kerenidis, and Dániel Szilágyi. “An optimal linear-

- combination-of-unitaries-based quantum linear system solver”. *ACM Transactions on Quantum Computing* **5**, 1–23 (2024).
- [58] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Quantum Random Access Memory”. *Physical Review Letters* **100**, 160501 (2008).
 - [59] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Architectures for a quantum random access memory”. *Physical Review A* **78**, 052310 (2008).
 - [60] Samuel Jaques and Arthur G Rattew. “QRAM: A Survey and Critique”. *Quantum* **9**, 1922 (2025).
 - [61] Patrick Rall. “Quantum algorithms for estimating physical quantities using block encodings”. *Physical Review A* **102**, 022408 (2020).
 - [62] Afonso Bandeira, Amit Singer, and Thomas Strohmer. “Mathematics of data science”. Draft available online. (2020). url: <https://people.math.ethz.ch/abandeira/BandeiraSingerStrohmer-MDS-draft.pdf>.
 - [63] James W. Demmel. “Applied numerical linear algebra”. *SIAM*. (1997).
 - [64] Zhaojun Bai, James Demmel, and Ming Gu. “An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems”. *Numerische Mathematik* **76**, 279–308 (1997).
 - [65] Iordanis Kerenidis and Anupam Prakash. “Quantum gradient descent for linear systems and least squares”. *Physical Review A* **101**, 022316 (2020).
 - [66] Lov Grover and Terry Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions” (2002).
 - [67] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. “Transformation of quantum states using uniformly controlled rotations”. *Quantum Information & Computation* **5**, 467–473 (2005). url: <https://dl.acm.org/doi/10.5555/2011670.2011675>.
 - [68] Ashwin Nayak and Felix Wu. “The quantum query complexity of approximating the median and related statistics”. In *Proc. ACM Symposium on Theory of Computing*. Pages 384–393. (1999).
 - [69] Scott Aaronson and Patrick Rall. “Quantum Approximate Counting, Simplified”. In *Proc. SIAM Symposium on Simplicity in Algorithms*. Pages 24–32. SIAM (2020).
 - [70] Scott Aaronson, Robin Kothari, William Kretschmer, and Justin Thaler. “Quantum Lower Bounds for Approximate Counting via Laurent Polynomials”. In *35th Computational Complexity Conference*. (2020).
 - [71] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. “Grand unification of quantum algorithms”. *PRX Quantum* **2**, 040203 (2021).
 - [72] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. “Explicit quantum circuits for block encodings of certain sparse matrices”. *SIAM Journal on Matrix Analysis and Applications* **45**, 801–827 (2024).
 - [73] Christoph Sünderhauf, Earl Campbell, and Joan Camps. “Block-encoding structured matrices for data input in quantum computing”. *Quantum* **8**, 1226 (2024).
 - [74] Daan Camps and Roel Van Beeumen. “FABLE: Fast approximate quantum circuits for block-encodings”. In *2022 IEEE International Conference on Quantum Computing and Engineering*. Pages 104–113. IEEE (2022).
 - [75] Kaoru Mizuta and Keisuke Fujii. “Recursive quantum eigenvalue and singular-value transformation: Analytic construction of matrix sign function by newton iteration”. *Physical Review Research* **6**, L012007 (2024).
 - [76] Guang Hao Low. “Quantum signal processing by single-qubit dynamics”. PhD thesis. Massachusetts Institute of Technology. (2017).

A Preliminaries on block-encodings and eigenvalue transformation

In this section we recall some basic definitions and known results for block-encodings. Block-encoding refers to the derivation of a unitary operator $\mathbf{U}_{\mathbf{A}}$ which can be described by a quantum circuit and “encodes” a given matrix $\mathbf{A}$, which was stated in Definition 1.1.

Constructing block-encodings with low cost is not a trivial task. A common approach in the literature is to assume a “black-box” unitary oracle that gives access to a (sparse) matrix $\mathbf{A}$. See eg. [49, 71]. Recent works take a step further, describing explicit quantum circuits for the block-encoding of special classes of matrices [72, 73, 50, 74]. In the main algorithm we assume the existence of a general subroutine that can prepare block-encodings, defined as follows.

Definition A.1. A subroutine $\mathbf{U}_{\mathbf{A}} \leftarrow \text{QENC}_{a,b}(\mathbf{A}, \epsilon)$ takes as input a matrix $\mathbf{A}$ of size N and an accuracy ϵ , and prepares an (a, b, ϵ) -QUBE of $\mathbf{A}$. a and b might be functions of N or $\|\mathbf{A}\|_{\xi}$, for some norm ξ .

It is straightforward to obtain a block-encoding after a diagonal shift.

Lemma A.1 (QSHIFT, follows from [49]). Let $h \in [-1/2, 1/2]$ be a scalar and $\mathbf{H}$ be a Hermitian matrix with $\|\mathbf{H}\| \leq 1/2$, and $\mathbf{U}_{\mathbf{H}}$ be an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}$ with cost $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$. We can construct a $(a + |h|, b + 1, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H} - h$ with a subroutine

$$\mathbf{U}_{\mathbf{H}-h} \leftarrow \text{QSHIFT}(\mathbf{U}_{\mathbf{H}}, h, a),$$

where $\mathcal{C}(\mathbf{U}_{\mathbf{H}-h}; \epsilon_{\text{enc}}) = \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}}) + O(1)$.

Proof. We use [49, Lemma 29] in a similar way as in [38]. First we construct a couple of 2-by-2 unitaries $\mathbf{P}_L |0\rangle = c_1 |0\rangle + c_2 |1\rangle$ and $\mathbf{P}_R |0\rangle = d_1 |0\rangle + d_2 |1\rangle$, as follows. Set $\beta = 1 + |h|/a$. The values c_1, d_1, c_2, d_2 are chosen such that $\beta(c_1^* d_1) = 1$ and $\beta(c_2^* d_2) = -h/a$. Let $\mathbf{W} = |0\rangle\langle 0| \otimes \mathbf{U}_{\mathbf{H}} + |1\rangle\langle 1| \otimes \mathbf{U}_{\mathbf{H}}$. One can verify that $(\mathbf{P}_L^\dagger \otimes \mathbf{I})\mathbf{W}(\mathbf{P}_R \otimes \mathbf{I})$ is an $(a + |h|, b + 1, \epsilon_{\text{enc}})$ -QUBE of $\mathbf{H} - h$. $\square$

A.1 Quantum eigenvalue transformation and matrix sign function

We can now describe how to prepare a block-encoding of the matrix sign function $\text{sgn}(\mathbf{H} - h)$ using Quantum Eigenvalue Transformation (QET). We follow the methodology of [38] which is convenient for our analysis, but we also note that many works have studied similar approximations [71, 49, 75, 76]. E.g., an elegant and practical approach is mentioned [75], using a recursive QET based on Padé approximations for the sign function, with sharp complexity bounds.

Corollary A.1 (Adapted from [38, 49]). Let $\mathbf{U}_{\mathbf{H}}$ be an $(a, b, \epsilon_{\text{enc}})$ -QUBE of a Hermitian matrix $\mathbf{H}$, $\|\mathbf{H}\| \leq 1/2$, with gate complexity $\mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$. Let h be a scalar in $[-1/2, 1/2]$, and assume that $\Lambda(\mathbf{H} - h) \subseteq [-1, -\Delta] \cup [\Delta, 1]$, for some (known) $\Delta \in (0, 1)$. Let $\epsilon_{\text{sgn}} \in (0, 1)$ be an input accuracy parameter. If $\epsilon_{\text{enc}} \leq \epsilon_{\text{sgn}} \frac{\Delta^2}{C_{\text{sgn}}(a+|h|)^2}$, for some absolute constant $C_{\text{sgn}} > 1$, then we can construct a unitary $\mathbf{U}_{\text{sgn}}$ which is an $(1, b + 2, \epsilon_{\text{sgn}})$ -QUBE of $\text{sgn}(\mathbf{H} - h)$. The cost of $\mathbf{U}_{\text{sgn}}$ is:

1. Queries to $\mathbf{U}_{\mathbf{H}-h}$: $O\left(\frac{a+|h|}{\Delta} \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right)\right)$, with $\mathcal{C}(\mathbf{U}_{\mathbf{H}-h}; \epsilon_{\text{enc}}) = \mathcal{C}(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}})$,
2. Number of qubits: $\log(N) + b + 2$,
3. Other elementary gates: $O\left(\frac{a+|h|}{\Delta} b \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right)\right)$

4. *Total complexity:* $C(\mathbf{U}_{\text{sgn}}; \Delta, \epsilon) = O\left((C(\mathbf{U}_{\mathbf{H}}; \epsilon_{\text{enc}}) + b) \cdot \frac{a+|h|}{\Delta} \log\left(\frac{1}{\epsilon_{\text{sgn}}}\right)\right)$.

Proof. Since $\mathbf{U}_{\mathbf{H}}$ is an $(a, b, \epsilon_{\text{enc}})$ -QUBE for $\mathbf{H}$, then, by definition, it is also an $(1, b, 0)$ -QUBE for a matrix $\tilde{\mathbf{H}}$, where $\|\mathbf{H} - a\tilde{\mathbf{H}}\| \leq \epsilon_{\text{enc}}$. We use Lemma A.1 to prepare an $(a+|h|, b+1, \epsilon_{\text{enc}})$ -QUBE $\mathbf{U}_{\mathbf{H}-h}$ for $\mathbf{H}-h$ with the same cost (up to constants). Again, by definition, $\mathbf{U}_{\mathbf{H}-h}$ is an $(1, b+1, 0)$ -QUBE for a matrix $\tilde{\mathbf{H}}'$ such that $\|(\mathbf{H}-h) - (a+|h|)\tilde{\mathbf{H}}'\| \leq \epsilon_{\text{enc}}$.

Using standard techniques from holomorphic functional calculus (see e.g. [19, Lemma B.1]) we know that as long as $\epsilon_{\text{enc}} \leq \frac{\epsilon_{\text{sgn}}}{2} \cdot \frac{\sigma_{\min}^2(\mathbf{H}-h)}{C_{\text{sgn}}(a+|h|)}$, for some constant C , then it holds that $\left\|\text{sgn}\left(\frac{\mathbf{H}-h}{a+|h|}\right) - \text{sgn}(\tilde{\mathbf{H}}')\right\| \leq \epsilon_{\text{sgn}}/2$. A sufficient upper bound for ϵ_{enc} is to set $\epsilon_{\text{enc}} \leq \epsilon_{\text{sgn}} \frac{\Delta^2}{2C_{\text{sgn}}(a+|h|)}$. It also holds that $\text{sgn}\left(\frac{\mathbf{H}-h}{a+|h|}\right) = \text{sgn}(\mathbf{H}-h)$ since sgn is scale invariant.

It remains to approximate $\text{sgn}(\tilde{\mathbf{H}}')$ with a polynomial (QET). From Weyl's inequality, $|\lambda_i(\mathbf{H}-h) - (a+|h|)\lambda_i(\tilde{\mathbf{H}}')| \leq \epsilon_{\text{enc}}$. Therefore, it follows that $\sigma_{\min}(\tilde{\mathbf{H}}') \geq \frac{\sigma_{\min}(\mathbf{H}-h) - \epsilon_{\text{enc}}}{a+|h|} \geq \frac{\Delta - \epsilon_{\text{enc}}}{a+|h|} \gg \frac{\Delta}{2(a+|h|)}$, where the last bound holds due to the aforementioned value of $\epsilon_{\text{enc}} \leq \epsilon_{\text{sgn}} \frac{\Delta^2}{2C_{\text{sgn}}(a+|h|)} \ll \Delta/2$. From [38, Lemma 3], we can find a polynomial $p(\cdot)$ with degree $l = O\left(\frac{2(a+|h|)}{\Delta} \log\left(\frac{2}{\epsilon_{\text{sgn}}}\right)\right)$, such that $|p(x)| \leq 1$ for all $|x| \leq 1$, and $|p(x) - \text{sgn}(x)| \leq \epsilon_{\text{sgn}}/2$ for all $x \in [-1, -\Delta/2(a+|h|)] \cup [\Delta/2(a+|h|), 1]$. Consequently, $\|p(\tilde{\mathbf{H}}') - \text{sgn}(\tilde{\mathbf{H}}')\| \leq \epsilon_{\text{sgn}}/2$.

From [49, Thm. 2] (or [38, Thm. 1]), we can obtain an $(1, b+2, 0)$ -QUBE of $p(\tilde{\mathbf{H}}')$ using l queries of $\mathbf{U}_{\mathbf{H}-h}$, $\mathbf{U}_{\mathbf{H}-h}^\dagger$, and $O((b+2)l)$ elementary gates. This is $\mathbf{U}_{\text{sgn}}$, and it is easy to see that it forms an $(1, b+2, \epsilon_{\text{sgn}})$ -QUBE for $\text{sgn}(\mathbf{H}-h)$, since:

$$\begin{aligned} \|p(\tilde{\mathbf{H}}') - \text{sgn}(\mathbf{H}-h)\| &= \|p(\tilde{\mathbf{H}}') - \text{sgn}\left(\frac{\mathbf{H}-h}{a+|h|}\right)\| \\ &\leq \|p(\tilde{\mathbf{H}}') - \text{sgn}(\tilde{\mathbf{H}}')\| + \|\text{sgn}(\tilde{\mathbf{H}}') - \text{sgn}\left(\frac{\mathbf{H}-h}{a+|h|}\right)\| \\ &\leq \epsilon_{\text{sgn}}/2 + \epsilon_{\text{sgn}}/2 \\ &= \epsilon_{\text{sgn}}. \end{aligned}$$

□

A.2 Computing the trace of a block-encoded matrix

The main algorithm requires a subroutine to compute the trace of a block-encoded matrix.

Definition A.2 ([61]). *Let ρ be a density operator on $\mathcal{H}_S$ and let $|\mathbf{0}\rangle$ be some easy-to-prepare state in $\mathcal{H}_S$. A unitary $\mathbf{U}_\rho$ on $\mathcal{H}_S \otimes \mathcal{H}_A$ is an R -preparation unitary of ρ if we have*

$$\rho = \text{tr}_{\mathcal{H}_A} (|\rho\rangle \langle \rho|) \quad (11)$$

where $|\rho\rangle = \mathbf{U}_\rho (|\mathbf{0}\rangle |\mathbf{0}\rangle)$ and $\mathbf{U}_\rho$ is implementable using R elementary gates.

Proposition A.1. *Let $\mathbf{I}_n$ denote the $2^n \times 2^n$ identity matrix. Then there exists a $2n$ -preparation of $\frac{1}{2^n} \mathbf{I}_n$ using n CNOT gates and n single-qubit gates on $\mathcal{H} \otimes \mathcal{H}$.*

Proof. Let $|i\rangle_n$ denote the i^{th} basis state on the 2^n -dimensional Hilbert space. Then we can prepare the state

$$|\rho\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n |i\rangle_n \quad (12)$$

by first applying $\mathbf{H}^{\otimes n} \otimes I_n$, where $\mathbf{H}$ denotes the Hadamard gate, and then CNOT gates between the j^{th} and the $(j+n)^{\text{th}}$ qubits for $j = 0, \dots, n-1$. Then, we have

$$\text{tr}_{\mathcal{H}}(|\rho\rangle\langle\rho|) = \text{tr}_{\mathcal{H}}\left(\frac{1}{2^n} \sum_{i,j=0}^{2^n-1} |i\rangle_n \langle i|_n |j\rangle_n \langle j|_n\right) = \frac{1}{2^n} \sum_{i,j=0}^{2^n-1} |i\rangle_n \langle j|_n \text{tr}_{\mathcal{H}}(|i\rangle_n \langle j|_n) \quad (13)$$

$$= \frac{1}{2^n} \sum_{i=0}^{2^n-1} |i\rangle_n \langle i|_n = \frac{1}{2^n} I_n. \quad (14)$$

□

Corollary A.2. *Let $\mathbf{A}$ be an $N \times N$ matrix, and $\delta_{\text{tr}}, \epsilon_{\text{tr}} \in (0, 1)$. Let $\mathbf{U}_{\mathbf{A}}$ be an $(a, b, \frac{\epsilon_{\text{tr}}}{2N})$ -QUBE of $\mathbf{A}$ with circuit complexity $\mathcal{C}(\mathbf{U}_{\mathbf{A}}; \frac{\epsilon_{\text{tr}}}{2N})$ such that*

$$\|\mathbf{A} - a\tilde{\mathbf{A}}\| \leq \frac{\epsilon_{\text{tr}}}{2N},$$

where $\tilde{\mathbf{A}} := (\langle 0|^{\otimes b} \otimes \mathbf{I})\mathbf{U}_{\mathbf{A}}(|0\rangle^{\otimes b} \otimes \mathbf{I})$. There exists a quantum algorithm that produces an estimate ξ of $\text{tr}(\mathbf{A})$ such that

$$|\xi - \text{tr}(\mathbf{A})| \leq \epsilon_{\text{tr}},$$

with probability at least $(1 - \delta_{\text{tr}})$. The algorithm has complexity:

1. Queries to $\mathbf{U}_{\mathbf{A}}$: $O\left(\frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta_{\text{tr}}}\right)\right)$,
2. Number of qubits: $2 \log(N) + b$,
3. Other elementary gates: $O\left(\log(N) \frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta_{\text{tr}}}\right)\right)$
4. Total complexity: $O\left((\log(N) + \mathcal{C}(\mathbf{U}_{\mathbf{A}}; \frac{\epsilon_{\text{tr}}}{2N})) \frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta_{\text{tr}}}\right)\right)$.

Proof. From Proposition A.1, there exists an R -preparation of $\rho = \frac{1}{N} I_n$ with cost $R = O(\log(N))$ elementary gates. From Lemma 5 of [61], we know that if $\mathbf{U}_{\mathbf{A}}$ is an $(a, b, 0)$ -QUBE of $\mathbf{A}$ with circuit complexity $\mathcal{C}(\mathbf{U}_{\mathbf{A}})$, and ρ has an R -preparation unitary, then for every $\epsilon, \delta > 0$ there exists an algorithm that produces an estimate ξ of $\text{tr}(\rho\mathbf{A})$ such that

$$|\xi - \text{tr}(\rho\mathbf{A})| \leq \epsilon \quad (15)$$

with probability at least $(1 - \delta)$. The algorithm uses $O\left(\frac{a}{\epsilon} \log\left(\frac{1}{\delta}\right)\right)$ applications of a Grover operator that requires four uses of the R -preparation unitary and two queries to $\mathbf{U}_{\mathbf{A}}$. Note that $\mathbf{U}_{\mathbf{A}}$ is an $(1, b, 0)$ -QUBE of $\tilde{\mathbf{A}}$. Using [61, Lemma 5] on $\tilde{\mathbf{A}}$ with error $\epsilon := \frac{\epsilon_{\text{tr}}}{2aN}$, we can compute an estimate ξ' such that

$$\left|\xi' - \text{tr}\left(\frac{1}{N}\tilde{\mathbf{A}}\right)\right| \leq \epsilon,$$

with probability at least $1 - \delta_{\text{tr}}$ and circuit complexity

$$O\left((\log(N) + \mathcal{C}(\mathbf{U}_{\mathbf{A}})) \frac{1}{\epsilon} \log\left(\frac{1}{\delta_{\text{tr}}}\right)\right) = O\left((\log(N) + \mathcal{C}(\mathbf{U}_{\mathbf{A}})) \frac{aN}{\epsilon_{\text{tr}}} \log\left(\frac{1}{\delta_{\text{tr}}}\right)\right).$$

Then we can set $\xi = aN\xi'$, which satisfies

$$|\xi - \text{tr}(\mathbf{A})| \leq \left|\xi - \text{tr}(a\tilde{\mathbf{A}})\right| + \left|\text{tr}(a\tilde{\mathbf{A}}) - \text{tr}(\mathbf{A})\right|$$

$$\begin{aligned}
&= aN \left| \xi' - \text{tr} \left(\frac{1}{N} \tilde{\mathbf{A}} \right) \right| + \left| \text{tr} (a \tilde{\mathbf{A}}) - \text{tr}(\mathbf{A}) \right| \\
&\leq aN \frac{\epsilon_{\text{tr}}}{2aN} + N \left\| a \tilde{\mathbf{A}} - \mathbf{A} \right\| \\
&\leq \epsilon_{\text{tr}}.
\end{aligned}$$

□