Faster Coherent Quantum Algorithms for Phase, Energy, and Amplitude Estimation

Patrick Rall

Quantum Information Center, University of Texas at Austin

We consider performing phase estimation under the following conditions: we are given only one copy of the input state, the input state does not have to be an eigenstate of the unitary, and the state must not be measured. Most quantum estimation algorithms make assumptions that make them unsuitable for this 'coherent' setting, leaving only the textbook approach. We present novel algorithms for phase, energy, and amplitude estimation that are both conceptually and computationally simpler than the textbook method, featuring both a smaller query complexity and ancilla footprint. They do not require a quantum Fourier transform, and they do not require a quantum sorting network to compute the median of several estimates. Instead, they use block-encoding techniques to compute the estimate one bit at a time, performing all amplification via singular value transformation. These improved subroutines accelerate the performance of quantum Metropolis sampling and quantum Bayesian inference.

Phase estimation is one of the most widely used quantum subroutines, because it grants quantum computers two unique capabilities. First, it yields a quadratic speedup in the accuracy of Monte Carlo estimates [BHMT00, Mon15, HM18]. Achieving 'Heisenberg limited' accuracy scaling has numerous applications in physics, chemistry, machine learning, and finance [Wright&20, ESP20, Rall20, An&20]. Second, it allows quantum computers to diagonalize unitaries in a certain restricted sense: if $U = \sum_j e^{2\pi i \lambda_j} |\psi_j\rangle \langle\psi_j|$ then phase estimation performs the transformation

$$\sum_j \alpha_j |0^n\rangle |\psi_j\rangle \to \sum_j \alpha_j |\lambda_j\rangle |\psi_j\rangle \tag{1}$$

where $|\lambda_j\rangle$ is an n-bit estimate of λ_j. This access to spectral information enables quantum speedups for linear algebra [HHL08, CKS15], studying physical systems [Temme&09, YA11, Lemi&19, JKKA20], estimating partition functions [Mon15], and performing Bayesian inference [HW19, AHNTW20].

Phase estimation is also a very complicated algorithm. Textbook phase estimation [NC00] requires the Quantum Fourier Transform (QFT), a tool we commonly associate with the exponential speedup of Shor's algorithm [Shor95]. However, phase estimation only delivers a quadratic speedup for estimation and the exponential speedup for linear algebra can sidestep the QFT [CKS15, GSLW18]. When applied to energies, phase estimation also requires Hamiltonian simulation, a quantum subroutine that is an entire subject of study in its own right: it requires recent innovations to apply optimally in a black-box setting [BCK15, LC1606, LC1610, LC17, GSLW18] and optimal Hamiltonian simulation for specific systems is still being actively studied [SHC20, Cam20]. Furthermore, phase estimation demands median amplification to guarantee an accurate answer. This can be challenging to implement coherently on a quantum computer [HNS01, Klauck02, Beals&12], because it requires many ancillae and a quantum sorting network. The probability with which phase estimation gives the correct answer (sometimes referred to as the 'Fejer kernel' [Rogg20]) is not always high enough to be amplified, which must be dealt with either by rounding or adaptivity [AA16]. The conceptual complexity of textbook phase estimation and the resulting computational overhead motivates a search for alternatives.

Fortunately, a lot of simplification is possible if we allow for 'incoherent' algorithms, where incoherence manifests via a variety of assumptions. We could be allowed to measure the quantum state, either because we are given many copies of the input state, or because we have the ability to prepare it inexpensively. Alternatively, we can assume a type of adaptivity where quantum states wait patiently without decohering while a classical computer performs a computation on the side. This assumption sometimes lets us repair the input state after using it [MW05, Temme&09, Poulin&17]. But most importantly, incoherent phase estimation algorithms usually require that the input state is an eigenstate of the unitary or Hamiltonian in question.

If enough of these assumptions hold, then 'iterative phase estimation' [Kit95] removes an enormous amount of conceptual and computational overhead. The estimate can be extracted one bit at a time, thereby removing the QFT. The accuracy of each bit can be amplified individually via many classical samples, removing the need for the quantum sorting network. The awkward notion of an 'n-bit estimate' can be removed and replaced with a traditional additive-error estimate [AA16]. Iterative phase estimation has seen many refinements and has been applied to gate set tomography and ground state energy estimation [Higgins07, KLY15, LT21]. An incoherent iterative approach also permits direct simplification of amplitude estimation [AR19, GGZW19].

However, for some applications of phase estimation maintaining coherence remains crucial. The original strategy for quantum matrix inversion [HHL08], quantum Metropolis sampling [Temme&09, YA11, Lemi&19], and a protocol for partition function estimation [Mon15], thermal state preparation, and Bayesian inference [HW19, AHNTW20] all violate the assumptions above. The eigenvalues must be estimated while preserving the superposition, and there is no guarantee that the input state is an eigenstate. These applications of 'coherent' phase estimation motivate the main question of this paper: does there exist a conceptually and computationally simpler algorithm for performing the transformation (1) while remaining coherent? That is: the input state is not necessarily an eigenstate, we are given exactly one copy, and there is no adaptive interaction with a classical computer.

In this paper we answer this question in the affirmative. We present simplified coherent algorithms for phase estimation, energy estimation, and amplitude estimation. None of these algorithms require a QFT, and they can be made arbitrarily close to the ideal transformation without a quantum sorting network to compute a median. These algorithms are about 14x to 20x faster than traditional phase estimation in terms of their query complexity, a performance metric that neglects the fact that they also require fewer ancilla qubits.

However, we also observe that there are unavoidable barriers to estimation in superposition. Consider an estimation algorithm that outputs a superposition of two estimates $\hat\lambda_j^{(1)}$ and $\hat\lambda_j^{(2)}$, both of which could be pretty close to the true value λ_j:

$$\sum_j \alpha_j |0^n\rangle |\psi_j\rangle \to \sum_j \alpha_j \left(\xi_j |\lambda_j^{(1)}\rangle + \zeta_j |\lambda_j^{(2)}\rangle \right) |\psi_j\rangle \tag{2}$$

However, it is a well known fact [BBBV97] that uncomputation cannot work in such a situation (unless one of ξ_j or ζ_j is ≈ 0). Thus, any algorithm that actually makes use of the estimates must necessarily damage the input superposition and can no longer be considered coherent.

Even worse, we show that *any unitary quantum algorithm* for estimation must perform a map in the form (2), and there always exist some values of λ_j where the neither of the ξ_j and ζ_j are ≈ 0. Therefore, the only way to get an algorithm that performs the deterministic transformation (1) is to assume certain values of λ_j do not appear. We refer to this as a **rounding promise,** and show that if the rounding promise holds, then our algorithms perform the map (1). However, we also construct our algorithms in such a way that they give reasonable non-deterministic estimates as in (2) even when no rounding promise holds.

In the following we give a brief outline of the method we use to construct the algorithms. They strongly resemble iterative phase estimation [Kit95], which works roughly like this: the estimate is computed one bit at a time, starting with the least significant bit. A 'Hadamard-test' computes each bit with a decent success probability, which is then amplified to a high probability by taking the majority vote of many samples. This process could be made coherent naively by computing each sample into an ancilla, but this requires so many ancillae that any performance benefit over the QFT- and median-based approach is lost. The key idea is to amplify without using any new ancillae.

To manipulate the probabilities of the Hadamard-test, we use 'block-encodings'. Block-encodings permit quantum computers to manipulate non-unitary matrices. In our case, the matrices' eigenvalues encode our probabilities. A unitary matrix U_A is a block-encoding of A if A is in the top-left corner:

$$U_A = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \tag{3}$$

Block-encoded matrices can be manipulated in two ways. First, *linear combinations of unitaries* [BCK15, CKS15] allow us to build block-encodings $\alpha A + \beta B$ given block-encodings of A and B. Linear combinations of unitaries allow us to make block-encodings of Hamiltonians presented as a sum of local terms, covering most practical applications. Second, *singular value transformation* [LC1610, LC1606, GSLW18] lets us apply certain polynomials $p(x)$ to the singular values of a block-encoded matrix A, using $\deg(p)$ many queries to controlled-U_A and its inverse.

Together, these two techniques permit the unification of many quantum algorithms into a single framework. For an accessible introduction to these methods, along with a comprehensive review of the most modern versions of these algorithms we refer to [MRTC21]. This work also presents the independent discovery of an algorithm very similar to our improved method of phase estimation, which is also sketched in [Chuang20].

To obtain the k'th bit ($0 \leq k < n-1$), iterative phase estimation performs a Hadamard-test on $U^{2^{n-k-1}}$, where $U = \sum_j e^{2\pi i \lambda_j} |\psi_j\rangle\langle\psi_j|$. The outcome of the test is a coin toss that is heads with probability $\cos^2(2^{n-k-1}\pi\lambda_i)$. We observe that the quantum circuit for the Hadamard-test resembles a linear combination of unitaries. Therefore, we can construct a block-encoding of a matrix whose eigenvalues encode the Hadamard-test probability for each eigenvector of U.

Iterative phase estimation then proceeds to perform the Hadamard-test several times and takes the majority vote. We observe that the probability that the majority vote is 1 is a polynomial in the Hadamard-test probability p:

$$\Pr[\text{majority vote is 1}] = \sum_{k=\lceil M/2 \rceil}^{M} \binom{M}{k} p^k (1-p)^{M-k} \tag{4}$$

We can therefore apply such an 'amplifying polynomial' [Diak09] directly to the block-encoding using singular value transformation, which requires no new ancillae. In fact, using techniques from [LC17], we can construct a polynomial that performs the same task with much smaller degree.

Now we have amplified the eigenvalues of the block-encoding to be close to 0 or 1, so we have a projector. To extract the 0/1-eigenvalue into a qubit we use the following novel technical tool:

Theorem. ***Block-measurement.*** *Say we have an approximate block-encoding of a projector* Π*. Then there exists a quantum channel that approximately implements the map:*

$$|0\rangle \otimes |\psi\rangle \to |1\rangle \otimes \Pi\, |\psi\rangle + |0\rangle \otimes (I - \Pi)\, |\psi\rangle \tag{5}$$

The channel is based on uncomputation [BBBV97], but the error analysis also features an interesting trick to deal with the case where the uncomputation fails. This theorem may find applications elsewhere.

Repeating the above procedure for each bit while carefully adjusting the phases at every step yields an algorithm that performs coherent phase estimation with no ancillae required. To estimate energies, we can construct a block-encoding with eigenvalues $\cos^2(2^{n-k-1}\pi\lambda_j)$ directly using the Jacobi-Anger expansion [GSLW18] rather than going through Hamiltonian simulation, which boosts the performance further, although now the algorithm does require ancillae. Finally, to perform coherent amplitude estimation, we construct a block-encoding of a 1x1 matrix containing the amplitude to be estimated and then invoke energy estimation.

A key research question of this paper is: Do these algorithms perform better than traditional phase estimation in practice? An asymptotic analysis is not sufficient to answer this question. Instead, we carefully bound the query complexity and failure probability of all algorithms involved using the diamond norm and carefully select constants to maximize the performance. Subsequently, we perform a numerical analysis of the query complexity. We find that we improve the query complexity of phase estimation by a factor of about 13x, and the query complexity of energy estimation is improved by about 20x. Our amplitude estimation algorithm inherits the speedup of the energy estimation algorithm.

This paper is structured as follows. In section 1 we carefully define the problem of coherent estimation and establish the notion of a rounding promise. Then we analyze the textbook method. In section 2 we present our novel algorithms for phase and energy estimation. In section 3 we discuss a numerical analysis of the query complexities of the algorithms. Then, in section 4 give a proof of the block-measurement theorem above and then show how to use energy estimation to perform non-destructive amplitude estimation in section 5.

1 Preliminaries

In this section we formally define the estimation tasks we want to solve (Definition 2). This requires setting up the notion of a 'rounding promise' (Definition 1) which highlights an inherent complication with coherent estimation that is not present in the incoherent case. We argue that this complication is unavoidable, and should be taken into account when coherent estimation is performed in practice. Next, we set up some simple technical tools for dealing with the error analysis of uncomputation (Lemmas 3,4). These will be used several times in this paper. Finally, we give a precise description and analysis of 'textbook' phase estimation (Proposition 5). Thus, this section clearly defines the problem and the previous state-of-the-art which we improve.

Our paper presents algorithms for phase, energy, and amplitude estimation. Amplitude estimation follows as a relativity simple corollary of energy estimation so we will only be talking about the prior two until Section 4. To talk about both phases and energies at once, we standardize input unitaries and Hamiltonians into the following form:

$$U = \sum_j e^{2\pi i \lambda_j} |\psi_j\rangle \langle\psi_j| \qquad\qquad H = \sum_j \lambda_j |\psi_j\rangle \langle\psi_j| \tag{6}$$

We refer to the $\{\lambda_j\}$ as the 'eigenvalues', and assume they live in the range $[0,1)$. While any unitary can be put into this form, the form places the constraint $0 \preceq H \prec I$ onto the Hamiltonian.

Our goal is to compute $|\lambda_j\rangle$ an 'n-bit estimate of λ_j'. In this paper, we take this to be an n-qubit register containing a binary encoding of the number floor$(2^n\lambda_j)$. Since $\lambda_j \in [0,1)$ we are guaranteed that floor$(2^n\lambda_j)$ is an integer $\in \{0, ..., 2^n - 1\}$ and thus has an n-bit encoding.

Consider phase estimation using a quantum circuit composed of elementary unitaries and controlled-$e^{2\pi i\lambda_j}$. Following an argument related to the polynomial method, we see that the resulting state must be of the form:

$$\sum_{x\in\{0,1\}^n} \sum_y \alpha_{x,y}(e^{2\pi i\lambda_j}) |x\rangle |\text{garbage}_{x,y}\rangle \tag{7}$$

where $\alpha_{x,y}(e^{2\pi i\lambda_j})$ is some polynomial of $e^{2\pi i\lambda_j}$. We would like $|x\rangle$ to encode floor$(2^n\lambda_j)$, meaning that $\alpha_{x,y}(e^{2\pi i\lambda_j}) = 1$ if $x = $ floor$(2^n\lambda_j)$ and $\alpha_{x,y}(e^{2\pi i\lambda_j}) = 0$ otherwise. This is impossible: $\alpha_{x,y}(e^{2\pi i\lambda_j})$ is a continuous function of λ_j, but the desired amplitude indicating $x = $ floor$(2^n\lambda_j)$ is discontinuous. For energy estimation a similar argument applies, just that the amplitudes are of the form $\alpha_{x,y}(\lambda_j)$. This argument even holds in the approximate case when we demand that $\alpha_{x,y} \leq \delta$ or $\geq 1-\delta$ for some small δ - the discontinuity is present regardless.

To some extent, this issue stems from the awkwardness of the notion of an 'n-bit estimate' since it requires rounding or flooring, a discontinuous operation, when only continuous manipulation of amplitudes is possible. A more comfortable notion is that of an additive-error estimate, used by [AA16, KP16] in their estimation algorithms.

However, the primary application of our algorithms is a situation where this simplification is not possible: Szegedy walks based on Hamiltonian eigenspaces [YA11, Lemi&19, JKKA20, WT21]. The original quantum Metropolis algorithm [Temme&09] implements a random walk over Hamiltonian eigenspaces, where the superposition is measured at every step and it is demonstrated that an additive error estimate is sufficient. But in order to harness a quadratic speedup due to quantum walks [Sze04], the measurement of energies must be made completely coherent, which is not achieved by an additive-error estimate unless the error is smaller than the gap between any eigenvalues. Therefore, we retain the notion of an 'n-bit estimate' in this paper. We could always fall back to an estimate with additive error $\varepsilon/2$ by computing $n = $ ceil$(\log_2(\varepsilon^{-1}))$ bits of accuracy.

However, the above argument still applies for additive-error estimation: the amplitude of any given estimate $|\hat\lambda\rangle$ is a continuous function of the eigenvalue λ_j. This means that the amplitude cannot be 0 or 1 everywhere, there must exist points where it crosses intermediate values in order to interpolate in between the two. In both the 'n-bit estimate' case and the additive-error case, this causes a problem for coherent quantum algorithms since the estimate cannot always be uncomputed. For some eigenvalues λ_j, the algorithm will yield an output state of the form $\alpha|\hat\lambda\rangle + \beta|\hat\lambda'\rangle$. Even if $\hat\lambda, \hat\lambda'$ are good estimates of λ_j, the uncompute trick [BBBV97] cannot be used for such an output state. Thus, the damage to the input superposition over $|\lambda_j\rangle$ is irreparable.

The only way to deal with this issue is to assume that the λ_j do not take certain values. Then the amplitudes can interpolate between 0 and 1 at those points. Observe that the discontinuities in the function $\text{floor}(2^n\lambda_j)$ occur at multiples of $1/2^n$. A 'rounding promise' simply disallows eigenvalues near these regions.

Definition 1. *Let $n \in \mathbb{Z}^+$ and $\alpha \in (0,1)$. A hermitian matrix H satisfies an (n,α)-**rounding promise** if it has an eigendecomposition $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$, all the eigenvalues λ_j satisfy $0 \leq \lambda_j < 1$, and for all $x \in \{0, ..., 2^n\}$:*

$$\lambda_j \notin \left[\frac{x}{2^n}, \frac{x}{2^n} + \frac{\alpha}{2^n}\right] \tag{8}$$

Similarly, a unitary matrix U satisfies an (n,α)-rounding promise if it can be written as $U = \sum_j e^{2\pi i \lambda_j} |\psi_j\rangle\langle\psi_j|$ and the phases λ_j satisfy the same assumptions.

We have $\lambda_j \in [0,1)$, and we have disallowed λ_j from certain sub-intervals of $[0,1)$. The definition above is chosen such that the total length of these disallowed subintervals is α, regardless of the value of n. We have essentially cut out an α-fraction of the allowed eigenvalues.

Guaranteeing a rounding promise demands an enormous amount of knowledge about the eigenvalues. We do not expect many Hamiltonians or unitaries in practice to actually provably satisfy such a promise. However, we expect that the estimation algorithms discussed in this paper will still perform pretty well even if they do not satisfy such a promise. One can increase the chances of success by setting α to be very small, so that the vast majority of the eigenvalues do not fall into a disallowed region. Then, if the input state is close to a uniform distribution over the eigenstates, then one can be fairly confident that only an α fraction of the eigenvalues in the support will be disallowed. The need for a rounding promise can be also sidestepped entirely by avoiding the use of energy estimation as a subroutine and approaching the desired problem directly. This has been accomplished for ground state finding [LT21].

The rounding promise is not just a requirement of our work in particular - it is a requirement for any coherent phase estimation protocol. The fact that coherent phase estimation does not work for certain phases is largely disregarded in the literature: for example, works like [KP16] neglect this issue entirely. However, there are some works that have observed this problem and attempt to mitigate it. Such methods are called 'consistent phase estimation' [TaShma13] since the error of their estimate is supposedly independent of the phase being estimated, thus allowing amplification to make the amplitudes always close to 0 or 1. We claim that all of these attempts fail, and furthermore that achieving coherent phase estimation without some kind of promise is impossible in principle. This is due to the polynomial method argument above: any coherent quantum algorithm's output state's amplitudes must be a continuous function of the phase, and continuous functions that are sometimes ≈ 0 and sometimes ≈ 1 must somewhere have an intermediate value. Recall that uncomputation only works when the amplitudes are close to 0 or 1. The error depends on if the phase is close to this transition point or not, so the error must depend on the phase. This issue was not taken into account by [Ambainis10], [KP17], and [KLLP18], since all of these either implicitly or explicitly state that there is an algorithm that approximately performs $|\psi_i\rangle|0^n\rangle \to |\psi_i\rangle|\lambda_i\rangle$ in superposition while λ_i

is a deterministic computational basis state[1]. A more promising approach is detailed in [TaShma13], which, crudely speaking, shifts the transition points by a classically chosen random amount. Now whether or not a phase is close to a transition point is independent of the phase itself, making amplification possible. [Ambainis10] describes a similar idea, calling it 'unique-answer' eigenvalue estimation. However, we claim that this only works for a single phase. If we consider, for example, a unitary whose phases are uniformly distributed in $[0, 1)$ at a sufficiently high density, then there will be a phase near a transition point for any choice of random shift. The only way to avoid the rounding promise is to sacrifice coherence and measure the output state.

All of the algorithms in this paper, including textbook phase estimation, achieve an asymptotic runtime of $O(2^n\alpha^{-1}\log(\delta^{-1}))$, where δ is the error in diamond norm. Before we move on to the formal definition of the estimation task, we informally argue that the α^{-1} dependence is optimal, via a reduction to approximate counting. We are given N items, K of which are marked. Following the standard method for approximate counting [BHMT00], we construct a Grover unitary whose phases λ_j encode $\arcsin(\sqrt{K/N})$. Given a $(1, \alpha)$-rounding promise, computing $\text{floor}(2^1\lambda_j)$ amounts to deciding if $\lambda_j \leq \frac{1-\alpha/2}{2}$ or $\lambda_j \geq \frac{1+\alpha/2}{2}$ given that one of these is the case. By shifting the λ_j around appropriately we can thus decide if $K \geq (1/2 + C\alpha)N$ or $K \leq (1/2 - C\alpha)N$, for some constant C obtained by linearising $\arcsin(\sqrt{K/N})$. We have achieved approximate counting with a promise gap $\sim \alpha$. Thus the $\Omega(\alpha^{-1})$ lower bound on approximate counting [NW98] implies our runtime must be $\Omega(\alpha^{-1})$.

Equipped with the notion of a rounding promise, we can define our estimation tasks. Many algorithms in this paper produce some kind of garbage, which can be dealt with the uncompute trick [BBBV97]. Rather than repeat the analysis of uncomputation in every single proof, we present a modular framework where we can deal with uncomputation separately. Furthermore, some applications may require computing some function of the final estimate, resulting in more garbage which also needs to be uncomputed. Rather than baking the uncomputation into each algorithm, it is thus more efficient to leave the decision of when to uncompute to the user of the subroutine.

Definition 2. *A* ***phase estimator*** *is a protocol that, given some $n \in \mathbb{Z}^+$ and $\alpha \in (0, 1)$, and any error target $\delta > 0$, produces a quantum circuit involving controlled U and $U^\dagger$ calls to some unitary U. If $U = \sum_j e^{2\pi i\lambda_j} |\psi_j\rangle\langle\psi_j|$ satisfies an (n, α)-rounding promise then this circuit implements a quantum channel that is δ-close in diamond norm to the map:*

$$|0^n\rangle|\psi_j\rangle \to |\mathit{floor}(\lambda_j 2^n)\rangle|\psi_j\rangle \tag{9}$$

Similarly, an ***energy estimator*** *is such a protocol that instead involves such calls to U_H which is a block-encoding (see Definition 9) of a Hamiltonian $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$ that satisfies an (n, α)-rounding promise.*

The ***query complexity of an estimator*** *is the number of calls to U or U_H in the resulting circuit, as a function of n, α, δ.*

*An estimator is said to '**have garbage**' or '**have** m **qubits of garbage**' if it is instead close to a map that produces some j-dependent m-qubit garbage state in another register:*

$$|0^n\rangle|0...0\rangle|\psi_j\rangle \to |\mathit{floor}(\lambda_j 2^n)\rangle|\mathit{garbage}_j\rangle|\psi_j\rangle \tag{10}$$

(Note that the quantum circuit can allocate and discard ancillae, but that does not count as garbage.)

[1][Ambainis10] makes use of such a map in Algorithm 4. [KP17] states this as Theorem II.2. [KLLP18] sketches but does not analyze a protocol for this after Claim 4.5.

*An estimator is said to '**have phases**' if the map introduces a j-dependent phase φ_j:*

$$|0^n\rangle |\psi_j\rangle \to e^{i\varphi_j} |floor(\lambda_j 2^n)\rangle |\psi_j\rangle \tag{11}$$

If an estimator is both 'with phases' and 'with garbage', then we can just absorb the $e^{i\varphi_j}$ into $|\text{garbage}_j\rangle$ so the 'with phases' is technically redundant. However, in our framework it makes more sense to treat phases and garbage independently since some of our algorithms are just 'with phases'.

The reason we measure errors in diamond norm has to do with uncomputation of approximate computations with garbage. Consider for example a transformation V acting on an answer register and a garbage register:

$$V|0\rangle|0...0\rangle = \sqrt{1-\varepsilon}\,|0\rangle|\text{garbage}_0\rangle + \sqrt{\varepsilon}\,|1\rangle|\text{garbage}_1\rangle \tag{12}$$

for some small nonzero ε. We copy the answer register into the output register:

$$\to \sqrt{1-\varepsilon}\,|0\rangle \otimes |0\rangle|\text{garbage}_0\rangle + \sqrt{\varepsilon}\,|1\rangle \otimes |1\rangle|\text{garbage}_1\rangle \tag{13}$$

and then we project the answer and garbage registers onto $V|0\rangle|0...0\rangle$:

$$\to \sqrt{1-\varepsilon}\,|0\rangle \cdot \sqrt{1-\varepsilon} + \sqrt{\varepsilon}\,|1\rangle \cdot \sqrt{\varepsilon} \tag{14}$$

The resulting state $(1-\varepsilon)|0\rangle + \varepsilon|1\rangle$ is not normalized, meaning that the projection $V|0\rangle|0...0\rangle$ succeeds with some probability < 1. Thus the uncomputed registers are not always returned to the $|0\rangle|0...0\rangle$ state.

At this point our options are either to postselect these registers to $|0\rangle|0...0\rangle$ or to discard them. Postselection improves the accuracy, but also implies that the algorithms do not always succeed. Since many applications of coherent energy estimation demand repeating this operation many times, it is important that the algorithm always succeeds. Thus, we need to discard qubits, so we need to talk about quantum channels. Therefore, the diamond norm is the appropriate choice for an error metric. Recall that the diamond norm is defined in terms of the trace norm [AKN98]:

$$|\Lambda|_\diamond := \sup_\rho |(\Lambda \otimes \mathcal{I})(\rho)|_1 \tag{15}$$

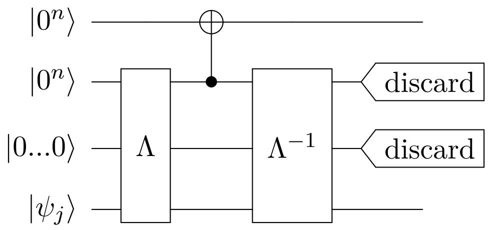
where $\mathcal{I}$ is the identity channel for some Hilbert space of higher dimension than Λ. The following analysis shows how to remove phases and garbage from an estimator, even in the approximate case.

Lemma 3. ***Getting rid of phases and garbage.*** *Given a phase/energy estimator with phases and/or garbage with query complexity $Q(n, \alpha, \delta)$ that is unitary, we can construct a phase/energy estimator without phases and without garbage with query complexity $2Q(n, \alpha, \delta/2)$.*

Proof. Without loss of generality, we assume we are given a quantum channel Λ that implements something close in diamond norm to the map:

$$|0^n\rangle|0...0\rangle|\psi_j\rangle \to e^{i\varphi_j}|\text{floor}(\lambda_j 2^n)\rangle|\text{garbage}_j\rangle|\psi_j\rangle \tag{16}$$

If the channel is actually without phases then we can just set $\varphi_j = 0$, and if it is actually without garbage then the following calculation will proceed without problems. Our strategy is just to use the uncompute trick, and then to discard the uncomputed ancillae. This requires us to implement Λ^{-1}, so we required that Λ be implementable by a unitary.



(17)

First, we consider the ideal case when Λ implements the map (16) exactly. If we use $|\psi_j\rangle$ as input and we stop the circuit before the discards, we produce a state $|\text{ideal}_j\rangle$:

$$\begin{aligned} |0^n\rangle\,|0^n\rangle\,|0...0\rangle\,|\psi_j\rangle &\to e^{i\varphi_j}\,|0^n\rangle\,|\text{floor}(\lambda_j 2^n)\rangle\,|\text{garbage}_j\rangle\,|\psi_j\rangle &(18)\\ &\to e^{i\varphi_j}\,|\text{floor}(\lambda_j 2^n)\rangle\,|\text{floor}(\lambda_j 2^n)\rangle\,|\text{garbage}_j\rangle\,|\psi_j\rangle &(19)\\ &\to |\text{floor}(\lambda_j 2^n)\rangle\,|0^n\rangle\,|0...0\rangle\,|\psi_j\rangle =: |\text{ideal}_j\rangle &(20)\end{aligned}$$

Let $\rho_{i,j}^{\text{ideal}} := |\text{ideal}_i\rangle\langle\text{ideal}_j|$, so that we can write down the ideal channel:

$$\Gamma_{\text{ideal}}(\sigma) := \sum_{i,j} \langle\psi_i|\,\sigma\,|\psi_j\rangle \cdot \rho_{i,j}^{\text{ideal}} \tag{21}$$

If we apply Λ with error $\delta/2$ instead we obtain the actual channel Γ such that:

$$\sup_\sigma |(\Gamma\otimes\mathcal{I})(\sigma) - (\Gamma_{\text{ideal}}\otimes\mathcal{I})(\sigma)|_1 \leq \delta \tag{22}$$

If A refers to the subsystems that start (and approximately end) in the states $|0^n\rangle\,|0...0\rangle$ then the final output state satisfies:

$$\begin{aligned} &\sup_\sigma |\text{Tr}_A\left((\Gamma\otimes\mathcal{I})(\sigma)\right) - \text{Tr}_A\left((\Gamma_{\text{ideal}}\otimes\mathcal{I})(\sigma)\right)|_1 &(23)\\ \leq &\sup_\sigma |\text{Tr}_A\left((\Gamma\otimes\mathcal{I})(\sigma) - (\Gamma_{\text{ideal}}\otimes\mathcal{I})(\sigma)\right)|_1 &(24)\\ \leq &\sup_\sigma |(\Gamma\otimes\mathcal{I})(\sigma) - (\Gamma_{\text{ideal}}\otimes\mathcal{I})(\sigma)|_1 \leq \delta &(25)\end{aligned}$$

where we have used the fact that for all ρ and subsystems A we have $|\text{Tr}_A(\rho)|_1 \leq |\rho|_1$. Upon plugging in $\sigma = |\psi_j\rangle\langle\psi_j|$ we can see that tracing out the middle register after applying Γ_{ideal} implements the map

$$|0^n\rangle\,|\psi_j\rangle \to |\text{floor}(\lambda_j 2^n)\rangle\,|\psi_j\rangle \tag{26}$$

so therefore the circuit in (17) is an estimator without phases and garbage as desired.

The proof is complete, up to the fact that $|\text{Tr}_A(\rho)|_1 \leq |\rho|_1$ for all ρ and subsystems A. Write ρ in terms of its eigendecomposition $\rho = \sum_i \lambda_i\,|\phi_i\rangle\langle\phi_i|$, and let $\rho_i := \text{Tr}_A(|\phi_i\rangle\langle\phi_i|)$:

$$|\text{Tr}_A(\rho)|_1 = \left|\text{Tr}_A\left(\sum_i \lambda_i\,|\phi_i\rangle\langle\phi_i|\right)\right|_1 = \left|\sum_i \lambda_i\rho_i\right|_1 \leq \sum_i |\lambda_i|\cdot|\rho_i|_1 = |\rho|_1 \tag{27}$$

□

This establishes that uncomputation works in the approximate case as expected. While we are reasoning about the diamond norm, we also present the following technical tool which will come in handy several times. In particular, we will need it for our analysis of textbook phase estimation.

Lemma 4. ***Diamond norm from spectral norm.*** *Say U, V are unitary matrices satisfying $|U - V| \leq \delta$. Then the channels $\Gamma_U(\rho) := U\rho U^\dagger$ and $\Gamma_V(\rho) := V\rho V^\dagger$ satisfy $|\Gamma_U - \Gamma_V|_\diamond \leq 2\delta$.*

Proof. We have that:

$$|\Gamma_U - \Gamma_V|_\diamond := \sup_\rho |(\Gamma_U \otimes \mathcal{I})(\rho) - (\Gamma_V \otimes \mathcal{I})(\rho)|_1 \tag{28}$$

$$= \sup_\rho \left|(U \otimes I)\rho(U \otimes I)^\dagger - (V \otimes I)\rho(V \otimes I)^\dagger\right|_1 \tag{29}$$

where $\mathcal{I}$ was the identity channel on some subsystem of dimension larger than that of U, V, and $|M|_1$ is the sum of the magnitudes of the singular values of M.

Let $\bar{U} = U \otimes I$ and $\bar{V} = V \otimes I$. Then:

$$\left|\bar{U} - \bar{V}\right| = |U \otimes I - V \otimes I| = |(U - V) \otimes I| = |U - V| \leq \delta \tag{30}$$

If we let $\bar{E} := \bar{U} - \bar{V}$ we can proceed with the upper bound:

$$|\Gamma_U - \Gamma_V|_\diamond = \sup_\rho \left|\bar{U}\rho\bar{U}^\dagger - \bar{V}\rho\bar{V}^\dagger\right|_1 \tag{31}$$

$$= \sup_\rho \left|\bar{U}\rho\bar{U}^\dagger - \bar{V}\rho\bar{V}^\dagger + \left(\bar{U}\rho V^\dagger - U\rho V^\dagger\right)\right|_1 \tag{32}$$

$$= \sup_\rho \left|\left(\bar{U}\rho\bar{U}^\dagger - U\rho V^\dagger\right) - \left(\bar{V}\rho\bar{V}^\dagger - \bar{U}\rho V^\dagger\right)\right|_1 \tag{33}$$

$$= \sup_\rho \left|\bar{U}\rho(\bar{U} - \bar{V})^\dagger + (\bar{U} - \bar{V})\rho\bar{V}^\dagger\right|_1 \tag{34}$$

$$\leq \delta + \delta \tag{35}$$

□

Now we have all the tools required to analyze the textbook algorithm [NC00]. This algorithm combines several applications of controlled-U with an inverse QFT to obtain the correct estimate with a decent probability. One can then improve the success probability via median amplification: if a single estimate is correct with probability $\geq \frac{1}{2} + \eta$ for some η then the median of $\lceil \ln(\delta^{-1})/2\eta^2 \rceil$ estimates is incorrect with probability $\leq \delta$.

However, median amplification alone is not sufficient to accomplish phase estimation as we have defined it in Definition 1. This is because any λ_i that is $\approx 10\% \cdot 2^{-n-1}$ close to a multiple of $1/2^n$, the probability of correctly obtaining floor$(2^n\lambda_j)$ is actually *less* than 1/2! This means that no matter how much median amplification is performed, this approach cannot achieve α below $\approx 10\%$. (For reference, a lower bound γ on the probability is plotted in Figure 1, although reading this figure demands some notation from the proof of the following proposition. We see that when $|\lambda_j^{(x)} - 1/2| \lesssim 10\%/2$ the probability of obtaining floor$(2^n\lambda_j)$ is less than 1/2.)

Furthermore, even if the signal obtained from the QFT could obtain arbitrarily small α, it would not achieve the desired α^{-1} scaling. This is because the amplification gap η scales linearly with α, and median amplification scales as η^{-2} due to the Chernoff-Hoeffding theorem. Therefore, we would obtain a scaling of α^{-2}.

Fortunately, achieving $\sim \alpha^{-1}$ is possible, using a different method for reducing α: we perform estimation for r additional bits which are then ignored. This makes use of the fact that α is independent of the number of estimated bits, but the gap away from the multiples of $1/2^n$, which is $\alpha/2^{n+1}$, is not. Every time we increase n, the width of each disallowed interval is chopped in half, but to compensate the number of disallowed intervals is doubled. If we simply round away some of the bits, then we do not care about the additional disallowed regions introduced. If we estimate and round r additional bits, we suppress α by a factor of 2^{-r} while multiplying our runtime by a factor of 2^r - so the scaling is $\sim \alpha^{-1}$.

We still need median amplification, though. In order to use the above strategy we need rounding to be successful: if an eigenvalue λ_j falls between two bins $\text{floor}(2^n\lambda_j)$ and $\text{floor}(2^n\lambda_j) + 1$, then it must be guaranteed to be rounded to one of those bins with failure probability at most δ. Before amplification, the probability that rounding succeeds is $\geq 8/\pi^2$, a constant gap above $1/2$ corresponding to $\alpha = 1/2$. We cannot guarantee a success probability higher than $8/\pi^2$ using the rounding trick alone, and thus need median amplification to finish the job.

Below, we split our analysis into two regimes: the $\alpha \leq 1/2$ regime and the $\alpha > 1/2$ regime. When $\alpha > 1/2$ then we do not really need the rounding trick described above, and we can achieve this accuracy with median amplification alone. Otherwise when $\alpha \leq 1/2$ we use median amplification to get to the point where rounding is likely to succeed, and then estimate r additional bits such that $\alpha 2^r \geq 1/2$.

Proposition 5. ***Standard phase estimation.*** *There exists an phase estimator with phases and garbage. Consider a unitary satisfying an (n, α)-rounding promise. Let:*

$$\gamma(x) := \frac{sin^2(\pi x)}{\pi^2 x^2} \qquad \eta_0 := \frac{8}{\pi^2} - \frac{1}{2} \qquad \delta_{med} := \frac{\delta^2}{6.25} \tag{36}$$

If $\alpha \leq 1/2$,let $r := \left\lceil \log_2\left(\frac{1}{2\alpha}\right)\right\rceil$. Then the phase estimator has query complexity:

$$(2^{n+r} - 1) \cdot \left\lceil \frac{\ln(\delta_{med}^{-1})}{2\eta_0^2} \right\rceil \tag{37}$$

and has $(n + r)\left\lceil \frac{\ln(\delta_{med}^{-1})}{2\eta_0^2} \right\rceil$ qubits of garbage.

Otherwise, if $\alpha > 1/2$, let $\eta := \gamma\left(\frac{1-\alpha}{2}\right) - \frac{1}{2}$. Then the phase estimator has query complexity:

$$(2^n - 1) \cdot \left\lceil \frac{\ln(\delta_{med}^{-1})}{2\eta^2} \right\rceil \tag{38}$$

and has $n\left\lceil \frac{\ln(\delta_{med}^{-1})}{2\eta^2} \right\rceil$ qubits of garbage.

Proof. We begin with the $\alpha > 1/2$ case which is simpler, and then extend to the $\alpha \leq 1/2$ case. The following is a review of phase estimation, which has been modified to estimate $\text{floor}(2^n\lambda_j)$ rather than $\text{round}\,(2^n\lambda_j)$:

1. Prepare a uniform superposition over times: $|+^n\rangle\,|\psi_j\rangle = \frac{1}{\sqrt{2^n}}\sum_{t=0}^{2^n-1}|t\rangle\,|\psi_j\rangle$.

2. Apply U to the $|\psi_j\rangle$ register t times, along with an additional phase shift of $\frac{-2\pi t(1-\alpha)}{2^{n+1}}$ to account for flooring:

$$\to \frac{1}{\sqrt{2^n}}\sum_{t=0}^{2^n-1}|t\rangle \otimes U^t e^{-\frac{2\pi i t(1-\alpha)}{2^{n+1}}}\,|\psi_j\rangle = \frac{1}{\sqrt{2^n}}\sum_{t=0}^{2^n-1} e^{2\pi i t\left(\lambda_j - \frac{1-\alpha}{2^{n+1}}\right)}\,|t\rangle\,|\psi_j\rangle \tag{39}$$

3. Apply an inverse quantum Fourier transform to the $|t\rangle$ register:

$$\to \frac{1}{2^n}\sum_{t=0}^{2^n-1}\sum_{x=0}^{2^n-1} e^{2\pi i t\left(\lambda_j - \frac{1-\alpha}{2^{n+1}}\right)} e^{-\frac{2\pi i}{2^n}tx} |x\rangle |\psi_j\rangle \tag{40}$$

$$= \sum_{x=0}^{2^n-1}\left[\frac{1}{2^n}\sum_{t=0}^{2^n-1} e^{2i\pi t\left(\lambda_j - \frac{x}{2^n} - \frac{1-\alpha}{2^{n+1}}\right)}\right] |x\rangle |\psi_j\rangle \tag{41}$$

$$= \sum_{x=0}^{2^n-1} \beta\left(\lambda_j^{(x)}\right) |x\rangle |\psi_j\rangle \tag{42}$$

where in the final line we have defined:

$$\lambda_j^{(x)} := 2^n\lambda_j - x - \frac{1-\alpha}{2} \tag{43}$$

$$\beta(\lambda_j^{(x)}) := \frac{1}{2^n}\sum_{t=0}^{2^n-1} e^{2\pi i t\lambda_j^{(x)}/2^n}. \tag{44}$$

4. Let η and δ_{med} be as in the theorem statement, and let:

$$M := \left\lceil \frac{\log(\delta_{\text{med}}^{-1})}{2\eta^2} \right\rceil \tag{45}$$

Repeat the above process for a total of M times. If we sum $\vec{x}$ over $\{0, ..., 2^n-1\}^M$, then we obtain:

$$\to \sum_{\vec{x}} \bigotimes_{l=1}^{M} \beta\left(\lambda_j^{(x_l)}\right) |k_l\rangle \otimes |\psi_j\rangle \tag{46}$$

5. Run a sorting network, e.g. [Beals&12], on the M estimates, and copy the median into the output register.

The query complexity analysis is straightforward: each estimate requires $2^n - 1$ applications of controlled-U, and there are M estimates. So all we must do is demonstrate accuracy. It is clear that the process above implements a map of the form:

$$|0^n\rangle |0...0\rangle |\psi_j\rangle \to \sum_{x=0}^{2^n-1} e^{i\varphi_{j,x}} \sqrt{p_{j,x}} |x\rangle |\text{gar}_{j,k}\rangle |\psi_j\rangle \tag{47}$$

for some probabilities $p_{j,x}$ and phases $\varphi_{j,x}$, since this is just a general description of a unitary map that leaves the $|\psi_j\rangle$ register intact. This way of writing the map lets us bound the error in diamond norm to the ideal map

$$|0^n\rangle |0...0\rangle |\psi_j\rangle \to e^{i\varphi_j} |\text{floor}(\lambda_j 2^n)\rangle |\text{gar}_j\rangle |\psi_j\rangle \tag{48}$$

(selecting $\varphi_j := \varphi_{j,\text{floor}(\lambda_j 2^n)}$ and $|\text{gar}_j\rangle := |\text{gar}_{j,\text{floor}(\lambda_j 2^n)}\rangle$) without uncomputing while still employing the standard analysis of phase estimation which just reasons about the probabilities $p_{j,x}$. These can be bounded from a median amplification analysis of the probabilities $\left|\beta\left(\lambda_j^{(x)}\right)\right|^2$. Consider a vector of M estimates $\vec{x}$. Then:

$$p_{j,x} := \sum_{\substack{\vec{x} \text{ where} \\ \text{median}(\vec{x})=x}} \prod_{l=1}^{M} \left|\beta(\lambda_j^{(x_l)})\right|^2 \tag{49}$$

First we show that probabilities $\left|\beta\left(\lambda_j^{(x)}\right)\right|^2$ associated with the individual estimates are bounded away from $\frac{1}{2}$. Using some identities we can rewrite:

$$|\beta(\lambda_j^{(x)})|^2 = \left|\frac{1}{2^n}\sum_{t=0}^{2^n-1} e^{2i\pi t\lambda_j^{(x)}/2^n}\right|^2 \tag{50}$$

$$= \left|\frac{1}{2^n}\frac{e^{2i\pi\lambda_j^{(x)}}-1}{e^{2i\pi\lambda_j^{(x)}/2^n}-1}\right|^2 \tag{51}$$

$$= \frac{1}{4^n}\frac{\sin^2(2\pi\lambda_j^{(x)}) + (\cos(2\pi\lambda_j^{(x)})-1)^2}{\sin^2(2\pi\lambda_j^{(x)}/2^n) + (\cos(2\pi\lambda_j^{(x)}/2^n)-1)^2} \tag{52}$$

$$= \frac{1}{4^n}\frac{2-2\cos(2\pi\lambda_j^{(x)})}{2-2\cos(2\pi\lambda_j^{(x)}/2^n)} \tag{53}$$

$$= \frac{1}{4^n}\frac{\sin^2(\pi\lambda_j^{(x)})}{\sin^2(\pi\lambda_j^{(x)}/2^n)} \tag{54}$$

Using the small angle approximation $\sin(\theta) \leq \theta$ for $0 \leq \theta \leq \pi$, we can derive:

$$|\beta(\lambda_j^{(x)})|^2 = \frac{1}{4^n}\frac{\sin^2(\pi\lambda_j^{(x)})}{\sin^2(\pi\lambda_j^{(x)}/2^n)} \geq \frac{1}{4^n}\frac{\sin^2(\pi\lambda_j^{(x)})}{\pi^2(\lambda_j^{(x)})^2/4^n} = \frac{\sin^2(\pi\lambda_j^{(x)})}{\pi^2(\lambda_j^{(x)})^2} =: \gamma(\lambda_j^{(x)}) \tag{55}$$

$\gamma(\lambda_j^{(x)})$ is a very tight lower bound to $|\beta(\lambda_j^{(x)})|^2$, and is plotted in Figure 1. We would like this probability to be larger than $\frac{1}{2}+\eta$ for some η when λ_j satisfies a rounding promise. This is guaranteed if we select:

$$\eta := \gamma\left(\frac{1-\alpha}{2}\right) - \frac{1}{2} \tag{56}$$

as in the theorem statement. From the figure, we see that when $\alpha > 1/2$ then $\eta > \eta_0$. This fact actually is not necessary for this construction to work, but observing this will be useful for the $\alpha \leq 1/2$ case. However, it is necessary to observe that when $\alpha > 1/2$ we are guaranteed that η is positive.

Then, from the rounding promise and the definition of $\lambda_j^{(x)}$, we are guaranteed that when $x = \text{floor}(2^n\lambda_j)$ then $|\beta(\lambda_j^{(x)})|^2 \geq 1/2+\eta$. Now we perform a standard median amplification analysis. The probability of a particular estimate x being 'correct' is $\geq 1/2+\eta$, so the probability of being incorrect is $\leq 1/2-\eta$. The only way that a median of M estimates can be incorrect is if more than half of the estimates are incorrect. Let X be the random variable counting the number of incorrect estimates. Then we can invoke the Chernoff-Hoeffding theorem:

$$\Pr[\text{median incorrect}] \leq \Pr\left[X \geq M/2\right] \tag{57}$$

$$\leq \Pr\left[X \geq M/2 + \mathbb{E}(X) - (1/2-\eta)M\right] \tag{58}$$

$$\leq \exp\left(-2M\eta^2\right) \tag{59}$$

We can bound the above by δ_{med}, a constant we will fix later, if we select $M := \left\lceil\frac{\log(\delta_{\text{med}}^{-1})}{2\eta^2}\right\rceil$ as we did above.

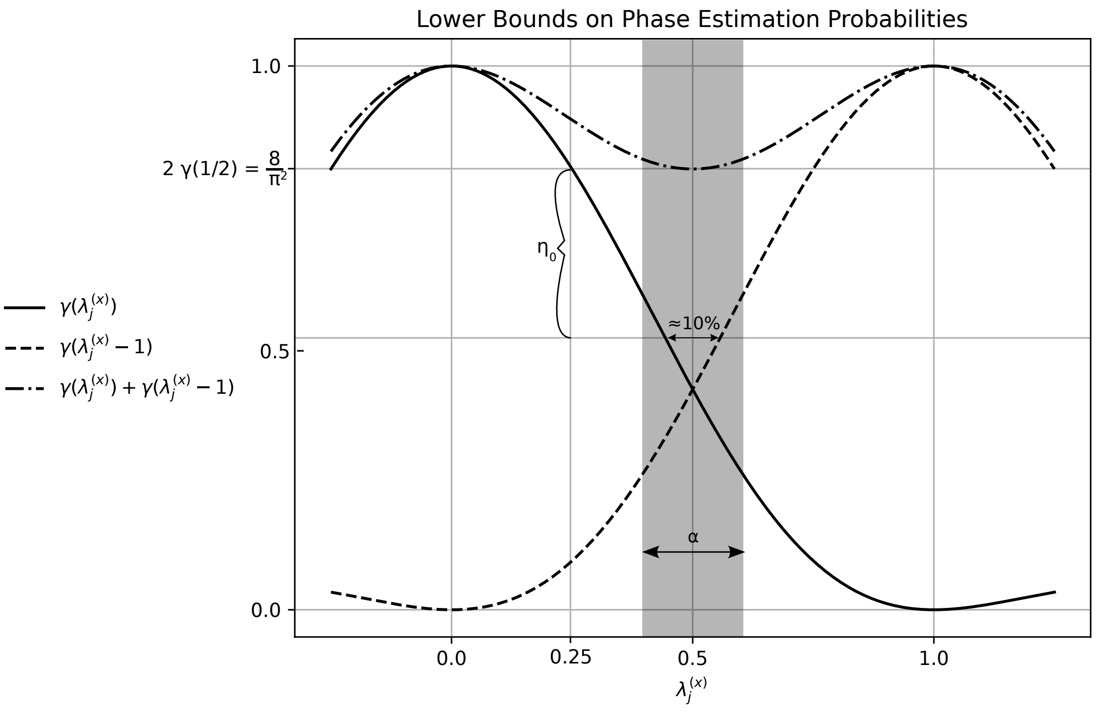


Figure 1: Sketch of $\gamma(\lambda_j^{(x)})$, a lower bound on the magnitude of the amplitude of phase estimation $|\beta(\lambda_j^{(x)})|^2$. Recall that the rounding promise guarantees that $\lambda_j \notin \left[\frac{x}{2^n}, \frac{x}{2^n}+\frac{\alpha}{2^n}\right]$ for all x, implying that $\lambda_j^{(x)} \notin [\frac{1-\alpha}{2}, \frac{1+\alpha}{2}]$ as indicated in the shaded region. From this sketch we draw two conclusions necessary for our proof. First, when $\alpha > 1/2$ then $\gamma(\lambda_j^{(x)})$ is guaranteed to be strictly greater than $1/2$, so $\eta > 0$ (in fact, $\eta > \eta_0$). Second, even when no rounding promise applies, the sum of the probabilities of two adjacent bins is at least $8/\pi^2$, which we use to define the amplification threshold η_0. Also visible in this figure is the impossibility of reducing α further than $\approx 10\%$ without rounding: several values of $\lambda_j^{(x)}$ near $1/2$ have probabilities less than $1/2$ and cannot be amplified.

We have demonstrated that $p_{\mathrm{j,x}} \geq 1-\delta_{\mathrm{med}}$ whenever $x = \mathrm{floor}(2^n\lambda_j)$. Now all that remains is a bound on the error in diamond norm. We begin by bounding the spectral norm. Let $p_j := p_{j,\mathrm{floor}(2^n\lambda_j)}$ and observe that $1 \geq p_j \geq 1-\delta_{\mathrm{med}}$. Taking the difference between equations (47) and (48) we obtain:

$$\left|(\sqrt{p_j}-1)e^{i\varphi_j}\,|\mathrm{floor}(\lambda_j 2^n)\rangle\,|\mathrm{gar}_j\rangle\,|\psi_j\rangle + \sum_{x,x\neq\mathrm{floor}(\lambda_j 2^n)} e^{i\varphi_{j,x}}\sqrt{p_{j,x}}\,|x\rangle\,|\mathrm{gar}_{j,x}\rangle\,|\psi_j\rangle\right| \tag{60}$$

$$= \sqrt{|(\sqrt{p_j}-1)e^{i\varphi_j}|^2 + \sum_{x,x\neq\mathrm{floor}(\lambda_j 2^n)} |e^{i\varphi_{j,x}}\sqrt{p_{j,x}}|^2} \tag{61}$$

$$= \sqrt{(\sqrt{p_j}-1)^2 + \sum_{x,x\neq\mathrm{floor}(\lambda_j 2^n)} p_{j_k}} \tag{62}$$

$$\leq \sqrt{(1-\sqrt{1-\delta_{\mathrm{med}}})^2 + \delta_{\mathrm{med}}} \tag{63}$$

$$\leq \sqrt{2-2\sqrt{1-\delta_{\mathrm{med}}} - \delta_{\mathrm{med}} + \delta_{\mathrm{med}}} \tag{64}$$

$$\leq \sqrt{2-2\sqrt{1-\delta_{\mathrm{med}}}} \tag{65}$$

Now we apply Lemma 4 to bound the diamond norm, and since the error is $\sim \sqrt{\delta_{\text{med}}}$ to leading order, we construct a bound that holds whenever $\sqrt{\delta_{\text{med}}} \leq 1$:

$$2\sqrt{2 - 2\sqrt{1-\delta_{\text{med}}}} \leq 2.5\sqrt{\delta_{\text{med}}} \tag{66}$$

If we select $\delta_{\text{med}} := \delta^2/6.25$ then the diamond norm is bounded by δ.

Having completed the $\alpha > 1/2$ case, we proceed to the $\alpha \leq 1/2$ case. As stated above, the construction we just gave actually also works when $\alpha \leq 1/2$. However, the asymptotic dependence on α is like $\sim \alpha^{-2}$ which is not optimal.

Looking again at Figure 1, we see that the sum of the probability of two adjacent bins is at least $8/\pi^2$, even when $\lambda_j^{(x)}$ is in a region disallowed by the rounding promise. Therefore, if we perform median amplification with gap parameter η_0 we are guaranteed that values of $\lambda_j^{(x)}$ that fall into a rounding gap will at least be rounded to an adjacent bin.

We can use this fact to achieve an $\sim \alpha^{-1}$ dependence. Recall that α is the fraction of the range $[0,1)$ where λ_j are not allowed to appear due to the rounding promise, independent of n. If we perform phase estimation with $n+1$ rather than n, then the width of each disallowed region is cut in half from $\alpha 2^{-n}$ to $\alpha 2^{-n-1}$ - but we also double the number of disallowed regions. However, if we simply ignore the final bit, then, because we are amplifying to η_0, any values of λ_j that fall into one of the newly introduced regions will simply be rounded. Thus, we cut the gaps in half without introducing any new gaps, so α is cut in half. We will make this argument more formal in a moment.

So, in order to achieve a particular α, we observe from Figure 1 that if we amplify to η_0 then the gaps have width $\frac{1}{2}\cdot 2^{-n}$. If we estimate an additional r bits, then the gaps have width $\frac{1}{2}\cdot 2^{-n-r}$. Solving $\frac{1}{2}\cdot 2^{-n-r} \leq \alpha 2^{-n}$ for r, we obtain

$$r := \left\lceil \log_2\left(\frac{1}{2\alpha}\right)\right\rceil. \tag{67}$$

We briefly make the n explicit in $\lambda_j^{(x)}$ by writing $\lambda_j^{(n,x)}$. After running the protocol at the beginning of the proof with $n+r$ bits, we obtain, for $\vec{x} \in \{0,...,2^n-1\}^M$, the state:

$$\sum_{\vec{x}} |\text{median}(\vec{x})\rangle \otimes \bigotimes_{l=1}^{M} \sum_{t=0}^{2^r-1} \beta(\lambda_j^{(n+r,\ 2^r x_l + t)})\, |2^r k_l + t\rangle \otimes |\psi_j\rangle \tag{68}$$

$$= \sum_{x=0}^{2^n-1} |x\rangle \otimes \sum_{\substack{\vec{x}\text{ where}\\ \text{median}(\vec{x})=kx}} \bigotimes_{l=1}^{M} \sum_{t=0}^{2^r-1} \beta(\lambda_j^{(n+r,\ 2^r x_l + t)})\, |2^r x_l + t\rangle \otimes |\psi_j\rangle \tag{69}$$

$$= \sum_{x=0}^{2^n-1} \sqrt{p_{j,x}} e^{i\varphi_{j,x}}\, |x\rangle \otimes |\text{gar}_{j,x}\rangle \otimes |\psi_j\rangle \tag{70}$$

where in the last step we have defined the normalized state $|\text{gar}_x\rangle$, the probability $p_{j,x}$, and the phase $\varphi_{j,x}$ via:

$$\sqrt{p_{j,x}} e^{i\varphi_{j,x}}\, |\text{gar}_{j,x}\rangle := \sum_{\substack{\vec{x}\text{ where}\\ \text{median}(\vec{x})=x}} \bigotimes_{l=1}^{M} \sum_{t=0}^{2^r-1} \beta(\lambda_j^{(n+r,\ 2^r x_l + t)})\, |2^r x_l + t\rangle \tag{71}$$

Now if we can demonstrate that $p_{j,x} \geq 1 - \delta_{\text{med}}$ when $x = \text{floor}(2^n \lambda_j)$ then the same argument as in (60-65) holds and we are done. Observe that:

$$p_{j,x} := \sum_{\substack{\vec{x} \text{ where} \\ \text{median}(\vec{x})=x}} \prod_{l=1}^{M} \sum_{t=0}^{2^r-1} \left| \beta(\lambda_j^{(n+r,\ 2^r x_l + t)}) \right|^2 \tag{72}$$

Say $x = \text{floor}(2^n \lambda_j)$. Then, consider $t := \text{floor}(2^{n+r}\lambda_j) - x2^r$, which is an integer $\in \{0, ..., 2^r - 1\}$. Then, looking at Figure 1, we see that:

$$|\beta(\lambda_j^{(n+r,\ 2^r x+t)})|^2 + |\beta(\lambda_j^{(n+r,\ 2^r x+t+1)})|^2 \geq \gamma(\lambda_j^{(n+r,\ 2^r x+t)}) + \gamma(\lambda_j^{(n+r,\ 2^r x+t)} - 1) \geq \frac{1}{2} + \eta_0 \tag{73}$$

Therefore the probability that the first n bits are x is $\sum_{t=0}^{2^r-1} \left| \beta(\lambda_j^{(n+r,\ 2^r x+t)}) \right|^2 \geq 1/2 + \eta_0$, which is all that was required to perform the median amplification argument above. So we can conclude that $p_{j,k} \geq 1 - \delta_{\text{med}}$, so the modified algorithm retains the same accuracy. □

2 Coherent Iterative Estimation

In this section we present the novel algorithms for phase estimation and energy estimation. As described in the introduction, both of these feature a strong similarity to 'iterative phase estimation' [Kit95], where the bits of the estimate are obtained one at a time. Unlike iterative phase estimation however, the state is never measured and the entire process is coherent. We therefore name these algorithms as 'coherent iterative estimators'.

Another similarity that these new algorithms share with the original iterative phase estimation is that the less significant bits are taken into account when obtaining the current bit. This greatly reduces the amount of amplification required for the later bits, so the runtime is vastly dominated by the estimation of the least significant bit. We will go into more detail on this later.

To permit discussion of coherent iterative estimation of phases and energies in a unified manner, we fit this idea into the modular framework of Definition 2 and Lemma 3. A 'coherent iterative estimator' obtains a single bit of the estimate, given access to all the previous bits. Several invocations of a coherent iterative estimator yield a regular estimator as in Definition 2. Furthermore, we can choose to uncompute the garbage at the very end using Lemma 3, or, as we will show, we can remove the garbage early, which prevents it from piling up.

Definition 6. *A* ***coherent iterative phase estimator*** *is a protocol that, given some n, α, any $k \in \{0, ..., n-1\}$, and any error target $\delta > 0$, produces a quantum circuit involving calls to controlled-U and $U^\dagger$. If the unitary U satisfies an (n, α)-rounding promise, then this circuit implements a quantum channel that is δ-close to some map that performs:*

$$|0\rangle |\Delta_k\rangle |\psi_j\rangle \to |bit_k(\lambda_j)\rangle |\Delta_k\rangle |\psi_j\rangle \tag{74}$$

Here $bit_k(\lambda_j)$ is the $(k+1)$'th least significant bit of an n-bit binary expansion, and $|\Delta_k\rangle$ is a k-qubit register encoding the k least significant bits. (For example, if $n = 4$ and $\lambda_j = 0.1011...$ then $bit_2(\lambda_j) = 0$ and $\Delta_2 = 11$.) Note that the target map is only constrained on a subspace of the input Hilbert space, and can be anything else on the rest.

An ***coherent iterative energy estimator*** *is the same thing, just with controlled-U_H and $U_H^\dagger$ queries to some block-encoding U_H of a Hamiltonian H that satisfies an (n, α)-rounding promise.*

A coherent iterative estimator can also be with garbage and/or with phases, just like in Definition 2.

Lemma 7. ***Stitching together coherent iterative estimators.*** *Given a coherent iterative phase/energy estimator with query complexity $Q(n, k', \alpha, \delta')$, we can construct a non-iterative phase/energy estimator with query complexity:*

$$\sum_{k=0}^{n-1} Q\left(n, k, \alpha, \delta \cdot 2^{-k-1}\right) \tag{75}$$

The non-iterative estimator has phases if and only if the coherent iterative estimator has phases, and if the coherent iterative estimator has m qubits of garbage then the iterative estimator has nm qubits of garbage.

Proof. We will combine n many coherent iterative phase/energy estimators, for $k = 0, 1, 2, .., n-1$. The diamond norm satisfies a triangle inequality so if we let the k'th iterative estimator have an error $\delta_k := \delta 2^{-k-1}$ then the overall error will be:

$$\sum_{k=0}^{n-1} \delta_k \leq \frac{\delta}{2} \sum_{k=0}^{n-1} 2^{-k} = \frac{\delta}{2}(2 - 2^{1-n}) \leq \delta \tag{76}$$

So now all that is left is to observe that the exact iterative estimators chain together correctly. This should be clear by observing that for all $k > 0$:

$$|\Delta_k\rangle = |\text{bit}_{k-1}(\lambda_j)\rangle \otimes ... \otimes |\text{bit}_0(\lambda_j)\rangle \tag{77}$$

and $|\Delta_0\rangle = 1 \in \mathbb{C}$ since when $k = 0$ there are no less significant bits. So the k'th iterative estimator takes the k least significant bits as input and computes one more bit, until finally at $k = n-1$ we have:

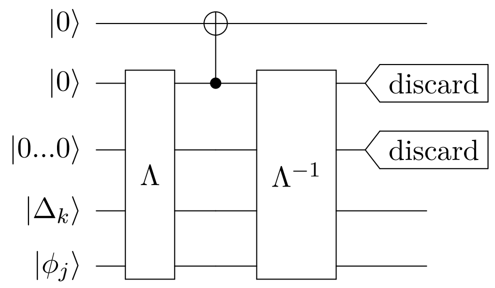
$$|\text{bit}_{n-1}(\lambda_j)\rangle\, |\Delta_{n-1}\rangle = |\text{bit}_{n-1}(\lambda_j)\rangle \otimes ... \otimes |\text{bit}_0(\lambda_j)\rangle = |\text{floor}(2^n \lambda_j)\rangle \tag{78}$$

The total query complexity is just the sum of the n invocations of the iterative estimators from $k = 0, ..., n-1$ with error δ_k.

If the iterative estimator has garbage, then the garbage from each of the n invocations just piles up. Similarly, if the estimator is with phases, and has an $e^{i\varphi_{j,k}}$ for the k'th invocation, then the composition of the maps will have a phase $\prod_{k=0}^{n-1} e^{i\varphi_{j,k}}$. □

Lemma 8. ***Removing garbage and phases from iterative estimators.*** *Given a coherent iterative phase/energy estimator with phases and/or garbage and with query complexity $Q(n, k, \alpha, \delta')$ that has a unitary implementation, we can construct a coherent iterative phase/energy estimator without phases and without garbage with query complexity $2Q(n, k, \alpha, \delta/2)$.*

Proof. This argument proceeds exactly the same as Lemma 3, just with the extra $|\Delta_k\rangle$ register trailing along. If Λ is the channel that implements the iterative estimator with garbage and/or phases, then we obtain an estimator without garbage and phases via:



(79)

□

The advantage of the modular framework we just presented is that maximizes the amount of flexibility when implementing these algorithms. How exactly uncomputation is performed will vary from application to application, and depending on the situation uncomputation may be performed before invoking Lemma 7 using Lemma 8, after Lemma 7 using Lemma 3, or not at all!

Of course, the idea of uncomputation combined with iterative estimation itself is quite simple, so given a complete understanding of the techniques we present the reader may be able to perform this modularization themselves. However, we found that this presentation significantly de-clutters the presentation of the main algorithms, those that actually implement the coherent iterative estimators from Definition 6. While the intuitive concept behind these strategies is not so complicated, the rigorous presentation and error analysis is quite intricate. We therefore prefer to discuss uncomputation separately.

2.1 Coherent Iterative Phase Estimation

This section describes our first novel algorithm, presented in Theorem 12. Before stating the algorithm in complete detail, performing an error analysis, and showing how to optimize the performance up to constant factors, we outline the tools that we will need and give an intuitive description.

As stated in the introduction, a very similar algorithm was independently discovered by [MRTC21]. The techniques of the two algorithms for phase estimation feature some minor differences: our work is more interested in maintaining coherence of the input state, the algorithm's constant-factor runtime improvement over prior art, and our runtime is $O(2^n)$ rather than $O(n2^n)$ ($O(n)$ vs $O(n\log n)$ respectively in the language of [MRTC21]). On the other hand, [MRTC21] elegantly show how a quantum Fourier transform emerges as a special case of the algorithm, and their presentation is significantly more accessible. Both methods avoid use of ancillae entirely.

A key tool for these algorithms is the block-encoding, which allows us to manipulate arbitrary non-unitary matrices. In this paper we simplify the notion a bit by restricting to square matrices with spectral norm ≤ 1.

Definition 9. *Say A is a square matrix $\in \mathcal{L}(\mathcal{H})$. A unitary matrix U_A is a* ***block-encoding*** *of A if U_A acts on m qubits and $\mathcal{H}$, and:*

$$(\langle 0^m| \otimes I)\, U\, (|0^m\rangle \otimes I) = A \tag{80}$$

We can also make m explicit by saying that A ***has a block-encoding with** m **ancillae**. We allow the quantum circuit implementing U to allocate ancillae in the $|0\rangle$ state and then to return them to the $|0\rangle$ state with probability 1. These ancillae are not postselected and do not contribute to the ancilla count m.*

A unitary matrix is a trivial block-encoding of itself. In this sense, we already have a block-encoding of the matrix:

$$U = \sum_j e^{2\pi i \lambda_j} |\psi_j\rangle \langle\psi_j| \tag{81}$$

Recall that the goal of the coherent iterative estimator is to compute $\mathrm{bit}_k(\lambda_j)$. The strategy involves preparing an approximate block-encoding of:

$$\sum_j \mathrm{bit}_k(\lambda_j) |\psi_j\rangle \langle\psi_j| \tag{82}$$

We begin by rewriting the above expression a bit. Recall that Δ_k is an integer from 0 to $2^{k+1}-1$ encoding the k less significant bits of λ_j. If we subtract $\Delta_k/2^n$ from λ_j, we obtain a multiple of $1/2^{n-k}$ plus something $< 1/2^n$ which we floor away. Then, $\text{bit}_k(\lambda_j)$ indicates if this is an even or an odd multiple. That means we can write:

$$\text{bit}_k(\lambda_j) = \text{parity}\left(\text{floor}\left(2^{n-k}\left(\lambda_j - \frac{\Delta_k}{2^n}\right)\right)\right) \tag{83}$$

Since $\lambda_j - \frac{\Delta_k}{2^n}$ is a multiple of $1/2^{n-k}$ plus something $< 1/2^n$, we equivalently have that $2^{n-k}\left(\lambda_j - \frac{\Delta_k}{2^n}\right)$ is an integer plus something less than $1/2^k$. For such values, we can write the parity(floor(x)) function in terms of a squared cosine that has been 'amplified':

$$\text{parity}(\text{floor}(x)) = \text{amp}\left(\cos^2\left(\frac{\pi}{2}[x+\phi]\right)\right) \tag{84}$$

$$\text{amp}(x) = \begin{cases} 1 \text{ if } x > 1/2 \\ 0 \text{ if } x < 1/2 \end{cases} \tag{85}$$

where the shift ϕ centers the extrema of the cosine in the intervals where x occurs. Therefore:

$$\begin{aligned} \text{bit}_k(\lambda_j) &= \text{amp}\left(\cos^2\left(\frac{\pi}{2}\left[2^{n-k}\left(\lambda_j - \frac{\Delta_k}{2^n}\right)+\phi\right]\right)\right) && (86)\\ &= \text{amp}\left(\cos^2\left(\pi\left[2^{n-k-1}\left(\lambda_j - \frac{\Delta_k}{2^n}\right)+\phi/2\right]\right)\right) && (87)\\ &= \text{amp}\left(\cos^2\left(\pi\lambda_j^{(k)}\right)\right) && (88)\end{aligned}$$

Where we have defined:

$$\lambda_j^{(k)} := 2^{n-k-1}\left(\lambda_j - \frac{\Delta_k}{2^n}\right) + \phi_k \tag{89}$$

for some k-dependent choice of phase $\phi_k = \phi/2$.

By applying a phase shift conditioned on the $|\Delta_k\rangle$ register, and then iterating it 2^{n-k-1} times, we can construct a block-encoding of the unitary with eigenvalues $\lambda_j^{(k)}$. Our goal is now to transform this unitary as follows:

$$\sum_j e^{2\pi i\lambda_j^{(k)}} |\psi_j\rangle\langle\psi_j| \quad \rightarrow \quad \sum_j \left[\text{amp}\left(\cos^2\left(\pi\lambda_j^{(k)}\right)\right)\right]|\psi_j\rangle\langle\psi_j| \tag{90}$$

To obtain a cosine use linear combinations of unitaries [BCK15, CKS15] to take a take a linear combination with the identity:

$$\begin{aligned} \sum_j \frac{e^{2\pi i\lambda_j^{(k)}}+1}{2}|\psi_j\rangle\langle\psi_j| &= \sum_j \cos\left(\pi\lambda_j^{(k)}\right)\left[e^{i\pi\lambda_j^{(k)}}|\psi_j\rangle\right]\langle\psi_j| && (91)\\ &= \sum_j \left|\cos\left(\pi\lambda_j^{(k)}\right)\right| \cdot \left[\pm e^{i\pi\lambda_j^{(k)}}|\psi_j\rangle\right]\langle\psi_j| && (92)\end{aligned}$$

where on the previous line $\pm$ indicates $\text{sign}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)$. This way the above is a singular value decomposition of the block-encoded matrix. So all that is left to do is to approximately transform the singular values a of the matrix above by:

$$a \rightarrow \text{amp}(a^2) \tag{93}$$

We will accomplish this using singular value transformation.

Lemma 10. ***Singular value transformation.*** *Say A is a square matrix with singular value decomposition $A = \sum_i a_i |\psi_i^l\rangle \langle\psi_i^r|$, and $p(x)$ is a degree-d even polynomial with real coefficients satisfying $|p(x)| \leq 1$ for $|x| \leq 1$, and suppose A has a block-encoding U_A. Then, for any $\delta > 0$, there exists a $O(poly(d, \log(1/\delta)))$ time classical algorithm that produces a circuit description of a block-encoding of the matrix:*

$$\tilde{p}(A) := \sum_i \tilde{p}(a_i) |\psi_i^r\rangle \langle\psi_i^r| \tag{94}$$

where $\tilde{p}(x)$ is a polynomial satisfying $|\tilde{p}(x) - p(x)| \leq \delta$ for $|x| \leq 1$. This block-encoding makes d queries to U_A or $U_A^\dagger$ (not controlled). If the block-encoding of A has m ancillae, then that of $\tilde{p}(A)$ has $m + 1$ ancillae. In the special case when $m = 1$ and the unitary implementing the block-encoding has the form $U_A = \sum_i V_i \otimes |\psi_i^l\rangle \langle\psi_i^r|$ where the V_i are qubit reflections, then it is possible to make the block-encoding of $\tilde{p}(A)$ also just have one ancilla.

Proof. This is a slightly simplified version of the main result of [GSLW18]. The $m > 1$ case involves an application of Corollary 18 which demands an extra control qubit that is also postselected. For an accessible review on the subject see [MRTC21].

We briefly elaborate on the $m = 1$ special case when $U_A = \sum_i V_i \otimes |\psi_i^l\rangle \langle\psi_i^r|$ and the V_i are reflections. We find that when this is the case, the $(W_x, S_z, \langle+| \cdot |+\rangle)$-QSP convention (see Theorem 13 of [MRTC21]) can actually be implemented by converting to the reflection convention and then using the circuit from Lemma 19 of [GSLW18], simply by applying a Hadamard gate to the ancilla register at the beginning and end of the circuit. This lets us implement polynomials with the constraints above without resorting to Corollary 18 of [GSLW18].

Note that the circuit from Lemma 19 of [GSLW18] (which is used in both the $m = 1$ and $m > 1$ cases) initializes an extra qubit to perform reflections. However, this qubit is guaranteed to be returned to the $|0\rangle$ state exactly and is not postselected, so it is not an ancilla in the sense of Definition 9. $\square$

Singular value transformation can perform the desired conversion if we can construct a polynomial $A(x)$ such that:

$$A(x) \approx \text{amp}(x) = \frac{1}{2} - \frac{1}{2}\text{sign}(2x - 1) \tag{95}$$

If we invoke the above lemma with the even polynomial $A(x^2)$ we get an approximation to the desired block-encoding of $\sum_j \text{bit}_k(\lambda_j) |\psi_j\rangle \langle\psi_j|$.

In particular, the behavior we must capture in the polynomial approximation is that $A(x) \approx 1$ when $x \in [0, 1/2 - \eta]$ and $A(x) \approx 0$ when $x \in [1/2 + \eta, 1]$ for some gap η away from $1/2$. If we view the input x as a probability, then $A(x)$ essentially 'amplifies' this probability to something close to 0 or 1 (and additionally flips the outcome). We hence call $A(x)$ an 'amplifying polynomial'.

One approach to constructing such an amplifying polynomial is to simply adapt a classical algorithm for amplification. We stated this method in the introduction: the polynomial is the probability that the majority vote of several coin tosses is heads, where each coin comes up heads with probability x. Then the desired properties can be obtained from the Chernoff-Hoeffding theorem [Diak09]. The number of coins we have to toss to accomplish a particular η, δ is the degree of the polynomial, which is bounded by $O(\eta^{-2} \log(\delta^{-1}))$.

However, this polynomial does not achieve the optimal η dependence of $O(\eta^{-1})$. This might be achieved by a polynomial inspired by a quantum algorithm for approximate counting, which does achieve the $O(\eta^{-1})$ dependence. But rather than go through such a complicated construction we simply adapt a polynomial approximation to the $\text{sign}(x)$ function developed in [LC17], which accomplishes the optimal $O(\eta^{-1}\log(\delta^{-1}))$.

Lemma 11. ***Quantum amplifying polynomial.*** *For any $0 < \eta, \delta < 1/2$, let:*

$$I_j(t) := \textit{the } j\textit{'th modified Bessel function of the first kind} \tag{96}$$

$$T_j(t) := \textit{the } j\textit{'th Chebyshev polynomial of the first kind} \tag{97}$$

$$k := \frac{\sqrt{2}}{4\eta}\sqrt{\ln\left(\frac{8}{\pi\delta}\right)} \tag{98}$$

For some $M_{\eta\to\delta}$, consider the polynomials:

$$p_{sgn}(x) := \frac{2ke^{-\frac{k^2}{2}}}{\sqrt{\pi}}\left(I_0\left(\frac{k^2}{2}\right)\cdot x + \sum_{j=1}^{\frac{1}{2}M_{(\eta\to\delta)}-\frac{1}{2}} I_j\left(\frac{k^2}{2}\right)(-1)^j\left(\frac{T_{2j+1}(x)}{2j+1} - \frac{T_{2j-1}(x)}{2j-1}\right)\right) \tag{99}$$

$$A_{\eta\to\delta}(x) := \frac{1}{2} - \frac{1}{2}\frac{p_{sgn}(2x-1)}{1+\delta/2} \tag{100}$$

Then there exists an $M_{\eta\to\delta} \in O\left(\eta^{-1}\log\left(\delta^{-1}\right)\right)$ such that $A_{\eta\to\delta}(x)$ is of degree $M_{\eta\to\delta}$ and satisfies the constraints:

$$\textit{if } 0 \le x \le 1 \textit{ then } 0 \le A_{\eta\to\delta}(x) \le 1 \tag{101}$$

$$\textit{if } 0 \le x \le \frac{1}{2} - \eta \textit{ then } A_{\eta\to\delta}(x) \ge 1 - \delta \tag{102}$$

$$\textit{if } \frac{1}{2} + \eta \le x \le 1 \textit{ then } A_{\eta\to\delta}(x) \le \delta \tag{103}$$

Proof. This is a more self-contained re-statement of Corollary 6 in Appendix A of [LC17], which constructs a polynomial $p_{\text{sgn},\kappa,\delta,n}(x) \approx \text{sign}(x)$ with various accuracy parameters. The polynomial above is $\frac{1}{2} - \frac{1}{2}p_{\text{sgn},\kappa,\delta/2,n}(2x-1)$, since after all we desired $A(x) \approx \frac{1}{2} - \frac{1}{2}\text{sign}(2x-1)$. To guarantee good behavior when $|x - 1/2| > \eta$, we select $\kappa := 4\eta$. The value of $M_{\eta\to\delta}$ itself can be computed by combining various results from that work's Appendix A. □

The method for obtaining $M_{\eta\to\delta}$ given η and δ is complicated enough that it is not worth re-stating here. However, the complexity is by all means worth it: in our numerical analyses we found that the polynomials presented in Appendix A of [LC17] feature excellent performance in terms of degree. This is a major source of the query complexity speedup of our algorithms.

The value of δ is chosen such that the final error in diamond norm is bounded. The value of η depends on how far away $\cos^2\left(\pi\lambda_j^{(k)}\right)$ is from $\frac{1}{2}$. Of course, there are several possible values of $\lambda_j^{(k)}$ where $\cos^2\left(\pi\lambda_j^{(k)}\right) = \frac{1}{2}$ exactly, so $\eta = 0$ and amplification is impossible. This is where the rounding promise comes in: it ensures that λ_Δ is always sufficiently far from such values, so that we can guarantee that $\cos^2\left(\pi\lambda_j^{(k)}\right)$ is always either $\ge \frac{1}{2} + \frac{\alpha}{2}$ or $\le \frac{1}{2} + \frac{\alpha}{2}$. So we select $\eta = \alpha/2$.

When $k = 0$ then indeed the rounding promise is the only thing guaranteeing that amplification will succeed. However, if the less significant bits have already been computed then the set of values that $\lambda_j^{(k)}$ can take is restricted. This is because the previous bits of λ_j

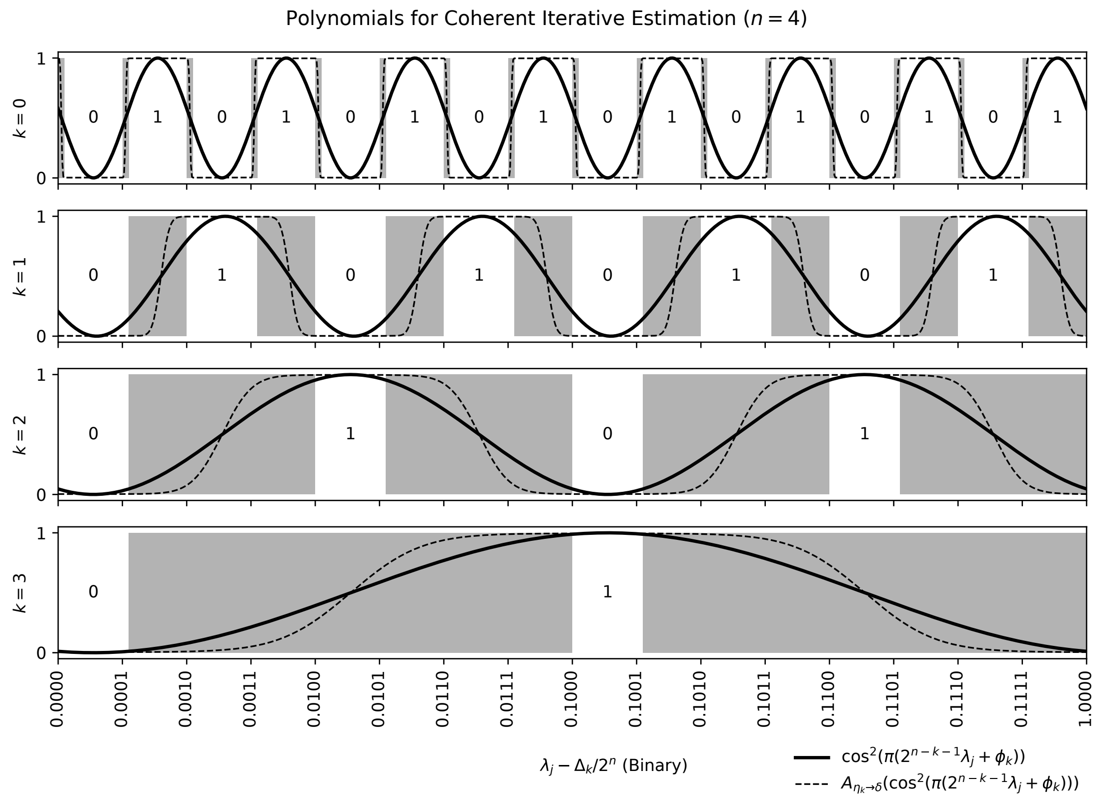


Figure 2: Sketch of the polynomials used in Theorems 12 and 15. In black we show a shifted $\cos^2(x)$ function that indicates if the bit is 0 or 1. Then, the amplifying polynomial from Lemma 11 is applied to it to yield the dashed line, which is either $\leq \delta$ or $\geq 1 - \delta$ depending on the bit. For the $k = 0$ bit, the gaps between the allowed intervals are guaranteed by the rounding promise. But as more bits are estimated and subtracted off, the gaps for $k \geq 1$ require no rounding promise, and also become larger and larger so less and less amplification is needed.

have been subtracted off. This widens the region around the solutions of $\cos^2\left(\pi\lambda_j^{(k)}\right) = \frac{1}{2}$ where λ_Δ cannot be found, allowing us to increase η. Furthermore, this can be done without relying on the rounding promise anymore: bits with $k \geq 1$ are guaranteed to be deterministic even if no rounding promise is present. That means that if the rounding promise is violated, then only the least significant bit can be wrong. The polynomials are sketched in Figure 2.

After constructing the amplifying polynomial $A_{\eta\to\delta}$, we use singular value transformation to apply $A_{\eta\to\delta}(x^2)$ which is even as required by Lemma 10. Now we have an approximate block-encoding of $\sum_j \text{bit}_k(\lambda_j) |\psi_j\rangle \langle\psi_j|$, which is in fact a projector. In the introduction we stated that we would use a block-measurement theorem to compute the map:

$$|0\rangle \otimes |\psi\rangle \to |1\rangle \otimes \Pi |\psi\rangle + |0\rangle \otimes (I - \Pi) |\psi\rangle \tag{104}$$

given an approximate block-encoding of Π. However, this general tool involves uncomputation which we specifically wanted to modularize. The fact that we are already measuring errors in terms of the diamond norm means that Lemmas 3 and 8 are already capable of dealing with garbage. We therefore defer the proof of this general tool to Section 4, specifically Theorem 19.

There is another reason to not use Theorem 19 in a black-box fashion, specifically for coherent iterative phase estimation. The block-encoding of $\sum_j \text{bit}_k(\lambda_j) |\psi_j\rangle \langle\psi_j|$ actually features only one ancilla qubit: the qubit we used to take the linear combination of U and the identity. That means that the block-encoding itself is already very close to the map $|0\rangle \otimes |\psi\rangle \to |0\rangle \otimes \Pi |\psi\rangle + |1\rangle \otimes (I - \Pi) |\psi\rangle$ (note the flipped output qubit). The details of this usage of the block-encoding will appear in the proof.

This completes the sketch of the procedure to implement the map

$$|0\rangle |\Delta_k\rangle |\psi_j\rangle \to e^{i\varphi_j} |\text{bit}_k(\lambda_j)\rangle |\Delta_k\rangle |\psi_j\rangle \tag{105}$$

for some phases φ_j. We can now state the protocol in detail, and perform the accuracy analysis.

Theorem 12. ***Coherent Iterative Phase Estimation.*** *There is a coherent iterative phase estimator with phases (and no garbage) and with query complexity:*

$$2^{n-k} \cdot M_{\eta \to \delta_{amp}} \tag{106}$$

where in the above $M_{\eta\to\delta}$ is as in Lemma 11, $\eta := \frac{1}{2} - \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)$ if $k \geq 1$ and $\eta := \frac{\alpha}{2}$ if $k = 0$, and δ_{amp} can be chosen to be $(1 - 10^{-m})\delta^2/24$ for any $m > 0$.

Proof. We construct the estimator as follows:

1. Construct a unitary that performs a phase shift depending on Δ_k - we call this $e^{-2\pi i\hat{\Delta}_k/2^n}$ employing some notation inspired by physics literature.

$$e^{-2\pi i\hat{\Delta}_k/2^n} := \sum_{\Delta_k=0}^{2^k-1} e^{-2\pi i\Delta/2^n} |\Delta_k\rangle\langle\Delta_k| = \bigotimes_{j=0}^{k-1} e^{-2\pi i\pi 2^{j-n}} \tag{107}$$

2. Rewrite the oracle unitary in this notation:

$$e^{2\pi i\hat{\lambda}} := U = \sum_j e^{2\pi i\lambda_j} |\psi_j\rangle\langle\psi_j| \tag{108}$$

   Then define:

$$\phi_0 := 1 - \text{mean}\left(\frac{1}{2} + \frac{\alpha}{2}, 1\right) \tag{109}$$

$$\phi_k := 1 - \text{mean}\left(\frac{1}{2}, \frac{1}{2} + \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)\right) \text{ if } k \geq 1 \tag{110}$$

$$\hat{\lambda}^{(k)} := 2^{n-k-1}\left(\hat{\lambda} - \frac{\hat{\Delta}_k}{2^n}\right) + \phi_k, \tag{111}$$

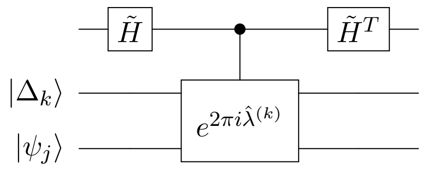
   and implement the corresponding phase shift:

$$e^{2\pi i\hat{\lambda}^{(k)}} := \left(e^{-2\pi i\hat{\Delta}_k/2^n} \otimes e^{2\pi i\hat{\lambda}}\right)^{2^{n-k-1}} \cdot e^{2\pi i\phi_k} \tag{112}$$

   This unitary acts jointly on the $|\Delta_k\rangle$ and $|\psi_j\rangle$ inputs. Since $k \in \{0, ..., n-1\}$, the exponent 2^{n-k-1} is always an integer.

3. Let $\tilde{H} := \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$ be a slightly modified Hadamard gate, and consider the following unitary, implemented via $e^{2\pi i \hat{\lambda}^{(k)}}$:

$$U_{\text{signal}}^{(k)} := \qquad (113)$$



Observe that:

$$U_{\text{signal}}^{(k)} = \sum_{\Delta_k}\sum_j \tilde{H}\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i \lambda_j^{(k)}} \end{bmatrix}\tilde{H}^T \otimes |\Delta_k\rangle\langle\Delta_k| \otimes |\psi_j\rangle\langle\psi_j| \tag{114}$$

$$= \sum_{\Delta_k}\sum_j e^{\pi i \lambda_j^{(k)}} \cdot \tilde{H}\begin{bmatrix} e^{-\pi i \lambda_j^{(k)}} & 0 \\ 0 & e^{\pi i \lambda_j^{(k)}} \end{bmatrix}\tilde{H}^T \otimes |\Delta_k\rangle\langle\Delta_k| \otimes |\psi_j\rangle\langle\psi_j| \tag{115}$$

$$= \sum_{\Delta_k}\sum_j e^{\pi i \lambda_j^{(k)}} \cdot \begin{bmatrix} \cos\left(\pi\lambda_j^{(k)}\right) & \sin\left(\pi\lambda_j^{(k)}\right) \\ \sin\left(\pi\lambda_j^{(k)}\right) & -\cos\left(\pi\lambda_j^{(k)}\right) \end{bmatrix} \otimes |\Delta_k\rangle\langle\Delta_k| \otimes |\psi_j\rangle\langle\psi_j| \tag{116}$$

In other words, $U_{\text{signal}}^{(k)}$ is a block-encoding of:

$$\sum_{\Delta_k}\sum_j \left|\cos\left(\pi\lambda_j^{(k)}\right)\right| \left[\pm e^{\pi i \lambda_j^k} |\Delta_k\rangle |\psi_j\rangle\right] [\langle\Delta_k| \langle\psi_j|] \tag{117}$$

The above is a singular value decomposition of the block-encoded matrix.

4. Choose the amplification threshold via:

$$\eta_k := \frac{1}{2} - \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right) \text{ if } k \geq 1 \tag{118}$$

$$\eta_0 := \frac{\alpha}{2} \tag{119}$$

Also, let $\delta_{\text{amp}} < 1$ be an error threshold we will pick later. Now, let $A_{\eta\to\delta_{\text{amp}}}(x)$ be the polynomial from Lemma 11. Viewing $U_{\text{signal}}^{(k)}$ as a block-encoding of $\cos\left(\pi\lambda_j^{(k)}\right)$, apply singular value transformation as in Lemma 10 to $U_{\text{signal}}^{(k)}$ with a polynomial $\tilde{p}(x)$ approximating $A_{\eta\to\delta_{\text{amp}}}(x^2)$ to accuracy δ_{svt}, which we also pick later.

Lemma 10 applies because $A_{\eta\to\delta_{\text{amp}}}(x^2)$ is even. Furthermore, $U_{\text{signal}}^{(k)}$ only has one ancilla qubit, and has the special form where the it implements a reflection on the ancilla (116). Thus, the circuit from Lemma 10 also just has one ancilla.

Call the resulting circuit $U_{\text{svt}}^{(k)}$, which implements the unitary:

$$U_{\text{svt}}^{(k)} = \sum_{\Delta_k}\sum_j \begin{bmatrix} \tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right) & \cdot \\ \gamma(\lambda_j^{(k)}) & \cdot \end{bmatrix} \otimes [|\Delta_k\rangle |\psi_j\rangle] [\langle\Delta_k| \langle\psi_j|] \tag{120}$$

for some matrix element $\gamma(\lambda_j^{(k)})$.

Now we prove that $U_{\text{svt}}^{(k)}$ is an iterative phase estimator with phases. It implements the map:

$$|0\rangle\,|\Delta_k\rangle\,|\psi_j\rangle \rightarrow \left(\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)|0\rangle + \gamma(\lambda_j^{(k)})\,|1\rangle\right)|\Delta_k\rangle\,|\psi_j\rangle \tag{121}$$

Note that $U_{\text{svt}}^{(k)}$ is a block-encoding with only one ancilla, and that ancilla is the output qubit of the map.

We must show that $U_{\text{svt}}^{(k)}$ is close in diamond norm to a map that leaves the first qubit as $|\text{bit}_k(\lambda_j)\rangle$ whenever Δ_k encodes the k least significant bits of an n-bit binary expansion of λ_j.

To study this map we will proceed through the recipe above, proving statements about the expressions encountered along the way. In step 2. we defined:

$$\lambda_j^{(k)} := 2^{n-k-1}\left(\lambda_j - \frac{\hat{\Delta}_k}{2^n}\right) + \phi_k, \tag{122}$$

We discuss the relationship between $\lambda_j^{(k)} - \phi_k$ and $\text{bit}_k(\lambda_j)$. If $k = 0$ then $\Delta_k = 0$, and due to the rounding promise we find λ_j in regions of the form $\frac{m}{2^n} + \left[\frac{\alpha}{2^n}, \frac{1}{2^n}\right]$ for integers m. Thus, we find $\lambda_j^{(0)} - \phi_0$ in regions of the form $\frac{m}{2} + \left[\frac{\alpha}{2}, \frac{1}{2}\right]$. The function $\text{bit}_k(\lambda_j)$ just indicates the parity of m. We can also write this as:

$$\text{bit}_0(\lambda_j) = \begin{cases} 0 & \text{if } \lambda_j^{(0)} - \phi_0 \in \text{floor}(\lambda_j^{(0)} - \phi_0) + [\frac{\alpha}{2}, \frac{1}{2}] \\ 1 & \text{if } \lambda_j^{(0)} - \phi_0 \in \text{floor}(\lambda_j^{(0)} - \phi_0) + [\frac{1}{2} + \frac{\alpha}{2}, 1] \end{cases} \tag{123}$$

If $k > 0$ then we also can show a similar property. While for $k = 0$ we used the rounding promise to guarantee that $\lambda_j^{(0)} - \phi_0$ only falls into certain regions, for larger k we simply use the fact that Δ_k has been subtracted off in the definition of $\lambda_j^{(k)}$. That means that the regions where $\text{bit}_{<k}(\lambda_j) = 1$ are no longer possible. We find:

$$\text{bit}_k(\lambda_j) = \begin{cases} 0 & \text{if } \lambda_j^{(k)} - \phi_k \in \text{floor}(\lambda_j^{(k)} - \phi_k) + \left[0, \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)\right] \\ 1 & \text{if } \lambda_j^{(k)} - \phi_k \in \text{floor}(\lambda_j^{(k)} - \phi_k) + \left[\frac{1}{2}, \frac{1}{2} + \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)\right] \end{cases} \quad \text{when } k > 0 \tag{124}$$

Note that the claim for $k > 0$ did not make use of the rounding promise, and is true regardless of if the rounding promise holds. These regions are shown in Figure 3.

In step 3. we defined U_{signal} which is a block-encoding of $\cos\left(\pi\lambda_j^{(k)}\right)$. Later, this will approximately be transformed by singular value transformation via $x \rightarrow A_{\eta\rightarrow\delta_{\text{amp}}}(x^2)$, so we employ a trigonometric identity:

$$\cos^2\left(\pi\lambda_j^{(k)}\right) = \frac{1 + \cos\left(2\pi\lambda_j^k\right)}{2} \tag{125}$$

Clearly this is a probability, and since cosine has period 2π, the $\text{floor}(\lambda_j^{(k)} - \phi_k)$ term does not matter. We argue that this probability is either $\geq 1/2 + \eta_k$ or $\leq 1/2 - \eta_k$ depending on $\text{bit}_k(\lambda_j)$. See Figure 3. The idea is that the ϕ_k are chosen precisely so that the troughs and peaks of $\cos^2\left(\pi\lambda_j^k\right)$ line up with the centers of the intervals corresponding to $\text{bit}_k(\lambda_j) = 0$ and $\text{bit}_k(\lambda_j) = 1$ respectively. Then, the nodes of $\cos^2\left(\pi\lambda_j^k\right)$ line up with the midpoints

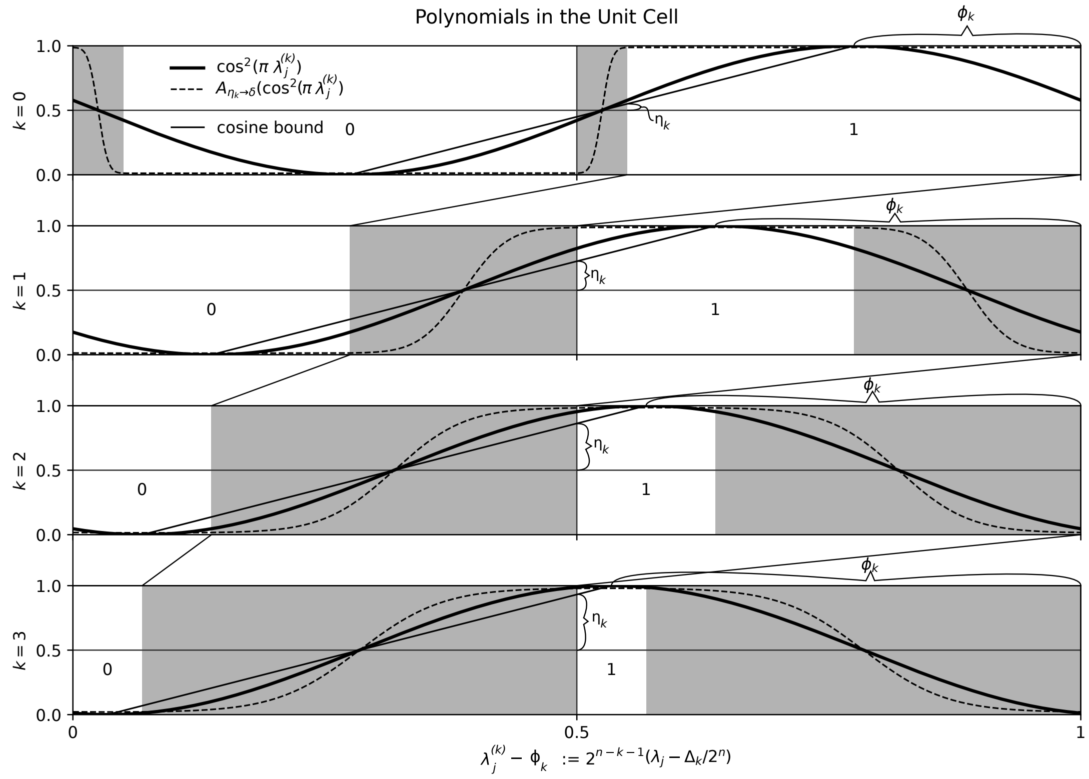


Figure 3: Sketch of the probability $\cos^2\left(\pi\lambda_j^{(k)}\right)$ which appears in the proof of Theorem 12.
We are guaranteed that $\lambda_j^{(k)}$ can only appear in the un-shaded regions: for $k = 0$ the rounding promise rules out the shaded regions, and for $k \geq 1$ we have subtracted $\Delta_k/2^n$ off of λ_j, preventing regions where previous bits are 1 .
We can see how the probability $\cos^2\left(\pi\lambda_j^{(k)}\right)$ is close to 1 if $\text{bit}_k(\lambda_j) = 0$ and close to 0 if $\text{bit}_k(\lambda_j) = 1$. To make this claim precise, we simply fit a line with slope 2 to the points where the probability intersects $1/2$, and see that this line alternatingly gives upper or lower bounds on the probability. So if $\eta_k/2$ is equal to half the distance between allowed intervals, then the probability is either $\geq 1/2 + \eta_k$ or $\leq 1/2 - \eta_k$ in the regions where $\lambda_j^{(k)}$ can appear.
The relationship of this figure to Figure 2 is fairly simple: the probability as a function of λ_j can be obtained by just tiling the 'unit cell' shown in this figure 2^{n-k-1} times. This also makes the ratio 2^{n-k-1} between λ_j and $\lambda_j^{(k)}$ intuitive.

of the gaps between the intervals. A line of slope 2 connecting a trough to a peak then forms an upper/lower bound on $\cos^2\left(\pi\lambda_j^k\right)$. If we select $\eta_k := \frac{1}{2} - \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)$ if and $\eta_0 := \frac{\alpha}{2}$ then these bounds show that $\cos^2\left(\pi\lambda_j^{(k)}\right)$ is alternatingly $\leq \frac{1}{2} - \eta_k$ and $\geq \frac{1}{2} + \eta_k$.

Then we have:

$$\cos^2(\pi\lambda_j^{(k)}) \text{ is } \begin{cases} \leq \frac{1}{2} + \eta_k & \text{if } \text{bit}_k(\lambda_j) = 0 \\ \geq \frac{1}{2} - \eta_k & \text{if } \text{bit}_k(\lambda_j) = 1 \end{cases} \tag{126}$$

By Lemma 11:

$$A_{\eta_k\to\delta_{\text{amp}}}\left(\cos^2(\pi\lambda_j^{(k)})\right) \text{ is } \begin{cases} \leq \delta_{\text{amp}} & \text{if } \text{bit}_k(\lambda_j) = 0 \\ \geq 1 - \delta_{\text{amp}} & \text{if } \text{bit}_k(\lambda_j) = 1 \end{cases} \tag{127}$$

And finally, since $\tilde{p}(x)$ approximates $A_{\eta_k \to \delta_{\text{amp}}}(x^2)$ to accuracy δ_{svt}:

$$\tilde{p}\left(\cos(\pi\lambda_j^{(k)})\right) \text{ is } \begin{cases} \leq \delta_{\text{amp}} + \delta_{\text{svt}} & \text{if } \text{bit}_k(\lambda_j) = 0 \\ \geq 1 - \delta_{\text{amp}} - \delta_{\text{svt}} & \text{if } \text{bit}_k(\lambda_j) = 1 \end{cases} \tag{128}$$

The circuit for U_{svt} is completely unitary, so the resulting state is normalized. Therefore:

$$\left|\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)\right|^2 + \left|\gamma(\lambda_j^{(k)})\right|^2 = 1 \tag{129}$$

Using this fact we can reason that if $\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right) \leq \delta_{\text{amp}} + \delta_{\text{svt}}$ then $|\gamma(\lambda_j^{(k)})|^2 \geq 1 - (\delta_{\text{amp}} + \delta_{\text{svt}})$.

Similarly, if $\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right) \geq 1 - \delta_{\text{amp}} - \delta_{\text{svt}}$, then:

$$|\gamma(\lambda_j^{(k)})|^2 \leq 1 - (1 - (\delta_{\text{amp}} + \delta_{\text{svt}}))^2 \leq 2(\delta_{\text{amp}} + \delta_{\text{svt}}) - (\delta_{\text{amp}} + \delta_{\text{svt}})^2 \tag{130}$$

Now that we have bounds on the amplitudes of the output state, we can bound its distance to $|\text{bit}_k(\lambda_j)\rangle$ for a favorable choice of $e^{i\varphi_j}$. Say $\text{bit}_k(\lambda_j) = 0$. Then we select $\varphi_j = 0$, so that:

$$\left|\left(\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)|0\rangle + \gamma(\lambda_j^{(k)})|1\rangle\right) - e^{i\varphi_j}|\text{bit}_k(\lambda_j)\rangle\right| \tag{131}$$

$$\leq \sqrt{\left|\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right) - 1\right|^2 + |\gamma(\lambda_\Delta)|^2} \tag{132}$$

$$\leq \sqrt{(\delta_{\text{amp}} + \delta_{\text{svt}})^2 + 2(\delta_{\text{amp}} + \delta_{\text{svt}}) - (\delta_{\text{amp}} + \delta_{\text{svt}})^2} \tag{133}$$

$$\leq \sqrt{2(\delta_{\text{amp}} + \delta_{\text{svt}})} \tag{134}$$

Otherwise, if $\text{bit}_k(\lambda_j) = 1$, then we define φ_j by $\gamma(\lambda_j^{(k)}) = e^{i\varphi_j}|\gamma(\lambda_j^{(k)})|$. That way:

$$\left|\left(\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)|0\rangle + \gamma(\lambda_j^{(k)})|1\rangle\right) - e^{i\varphi_j}|\text{bit}_k(\lambda_j)\rangle\right| \tag{135}$$

$$\leq \sqrt{\left|\tilde{p}\left(\cos\left(\pi\lambda_j^{(k)}\right)\right)\right|^2 + \left|e^{i\varphi_j}(|\gamma(\lambda_j^{(k)})| - 1)\right|^2} \tag{136}$$

$$\leq \sqrt{|\delta_{\text{amp}} + \delta_{\text{svt}}|^2 + |\delta_{\text{amp}} + \delta_{\text{svt}}|^2} \tag{137}$$

$$\leq \sqrt{2(\delta_{\text{amp}} + \delta_{\text{svt}})} \tag{138}$$

So either way, the output state is within $\sqrt{2(\delta_{\text{amp}} + \delta_{\text{svt}})}$ in spectral norm of that of the ideal state. Thus, the unitary map U_{svt} is close in spectral norm to some ideal unitary. Invoking Lemma 4, the distance in diamond norm is at most:

$$2\sqrt{2(\delta_{\text{amp}} + \delta_{\text{svt}})} \leq \delta \tag{139}$$

The inequality above holds if we select, for any $m > 0$:

$$\delta_{\text{amp}} := (1 - 10^{-m}) \cdot \frac{\delta^2}{8} \qquad \delta_{\text{svt}} := 10^{-m} \cdot \frac{\delta^2}{8} \tag{140}$$

This solution to the inequality relies on the fact that only δ_{amp} actually enters the query complexity, so if classical resources are cheap but query complexity is expensive then we can make the classical computer do as much work as possible by making m larger.

Finally, we compute the query complexity. An invocation of $e^{2\pi i \hat{\lambda}_j^{(k)}}$ requires 2^{n-k-1} invocations of $U = e^{2\pi i \hat{\lambda}}$. By Lemma 10, the number of queries made by the unitary U_{svt} to U_{signal} is the degree of the polynomial. By Lemma 11, the polynomial $A_{\eta\to\delta_{\text{amp}}}(x^2)$ has degree $2 \cdot M_{\eta\to\delta}$, so the query complexity is:

$$2^{n-k-1} \cdot 2 \cdot M_{\eta\to\delta_{\text{amp}}} \tag{141}$$

□

A really nice feature of the coherent iterative phase estimator we present is that it produces no garbage qubits. All singular value transformation is performed on the final output qubit. It does still produce extra phase shifts between the eigenstates, which in some applications may still need to be be uncomputed. However, in applications where phase differences between eigenstates do not matter, like thermal state preparation, we expect that this uncomputation step can be skipped.

To finish the discussion of coherent iterative phase estimation, we stitch the iterative estimator we just defined into a regular phase estimator. In doing so, we also remark on what happens when no rounding promise is present. A summary of the construction is presented in Figure 4.

Corollary 13. ***Improved phase estimation.*** *The coherent iterative phase estimator from Theorem 12 can be combined with Lemma 7 to make a phase estimator with phases with query complexity at most:*

$$O\left(2^n \alpha^{-1} \log(\delta^{-1})\right) \tag{142}$$

assuming α is bounded away from 1 by a constant.

Furthermore, if no rounding promise is given, then the estimator δ-approximates in diamond norm a map:

$$|0^n\rangle |\psi_j\rangle \to \left(\xi \,|\textit{floor}(2^n\lambda_j)\rangle + \zeta\, |\lambda_j'\rangle\right) |\psi_j\rangle \tag{143}$$

for some complex amplitudes ξ, ζ and $\lambda_j' = \textit{floor}(2^n\lambda_j) - 1 \bmod 2^n$ is an erroneous estimate. The performance is the same, except that $0 < \alpha < 1$ can be any constant.

Proof. Write $\eta_k := \frac{1}{2} - \frac{1}{2^k}\left(\frac{1}{2} + \frac{\alpha}{2}\right)$ if $k \geq 1$ and $\eta_0 := \frac{\alpha}{2}$ if $k = 0$, and, following Lemma 7, demand an accuracy of $\delta_{\text{amp},k} := (1 - 10^{-m})(\delta 2^{-k-1})^2/8$ for the k'th bit. Recall from Lemma 11 that $M_{\eta_k\to\delta_{\text{amp}}} \in O\left(\eta_k^{-1}\log(\delta_{\text{amp}}^{-1})\right)$. Then, the overall query complexity is:

$$\sum_{k=0}^{n-1} 2^{n-k} \cdot M_{\eta_k\to\delta_{\text{amp},k}} \in O\left(\sum_{k=0}^{n-1} 2^{n-k}\eta_k^{-1}\log(\delta_{\text{amp},k}^{-1})\right) \tag{144}$$

$$= O\left(\sum_{k=0}^{n-1} 2^{n-k}\eta_k^{-1}\log(2^{k+1}\delta^{-1})\right) \tag{145}$$

When $k = 0$ we have $\eta_0 = \frac{\alpha}{2}$, and when $k > 0$ we have $\eta_k > \frac{1-\alpha}{4}$ which is bounded from below by a constant. We can use this to split the sum:

$$\leq O\left(2^{n-0}\eta_0^{-1}\log(2^{0+1}\delta^{-1}) + \sum_{k=1}^{n-1} 2^{n-k}\eta_k^{-1}\log(2^{k+1}\delta^{-1})\right) \tag{146}$$

$$\leq O\left(2^{n}\alpha^{-1}\log(\delta^{-1}) + \sum_{k=1}^{n-1} 2^{k}(k+\log(\delta^{-1}))\right) \tag{147}$$

$$\leq O\left(2^{n}\alpha^{-1}\log(\delta^{-1}) + 2(2^n-n-1) + (2^n-2)\log(\delta^{-1})\right) \tag{148}$$

$$\leq O\left(2^{n}\alpha^{-1}\log(\delta^{-1})\right) \tag{149}$$

This completes the runtime analysis. Now we turn to the case when no rounding promise is present. Notice that when $k \geq 1$, the regions where λ_j is assumed not to appear are guaranteed by the previous estimators *definitely* outputting 1 for previous bits regardless of the promise. Thus, the only bit that can be wrong is the first bit. When an eigenvalue λ_j falls into a region disallowed by the rounding promise, the first bit will be some superposition $\xi\,|0\rangle + \zeta\,|1\rangle$.

Flipping the final bit of an estimate in general results in an error of ± 1. However, recall that when estimating future bits the value of $\Delta_k/2^n$ is subtracted off from λ_j. That means that when we erroneously measure a 1, the rest of the algorithm proceeds to measure $\lambda_j - 1/2^n$ instead. As a result, if the first bit is wrong, then the algorithm will output $\text{floor}(2^n\lambda_j) - 1$ instead. Since the algorithm is periodic in λ_j with period 1, the output will be $2^n - 1$ if an error occurs when $\text{floor}(2^n\lambda_j) = 0$.

□

Notice that if we had allocated the error evenly between the n steps, then we would have incurred an extra $O(2^n \log n)$ term in the above. Spreading the error via a geometric series avoids this, and we find that we obtain better constant factors with this choice as well. This is because the $k = 0$ term dominates, so we want to make $\delta_{\text{amp},k}$ as large as possible.

2.2 Coherent Iterative Energy Estimation

While for phase estimation we are given access to $\sum_j e^{2\pi i\lambda_j}\,|\psi_j\rangle\langle\psi_j|$, for energy estimation we have a block-encoding of $\sum_j \lambda_j\,|\psi_j\rangle\langle\psi_j|$. For phase estimation we could relatively easily synthesize a cosine in the eigenvalues, just by taking a linear combination with the identity. But for energy estimation we must build the cosine directly.

To do so we leverage a tool employed by [GSLW18] to perform Hamiltonian simulation. The Jacobi-Anger expansion yields highly efficient polynomial approximations to $\sin(tx)$ and $\cos(tx)$. To perform Hamiltonian simulation one then takes the linear combination $\cos(tx) + i\sin(tx)$. However, we only need the cosine component.

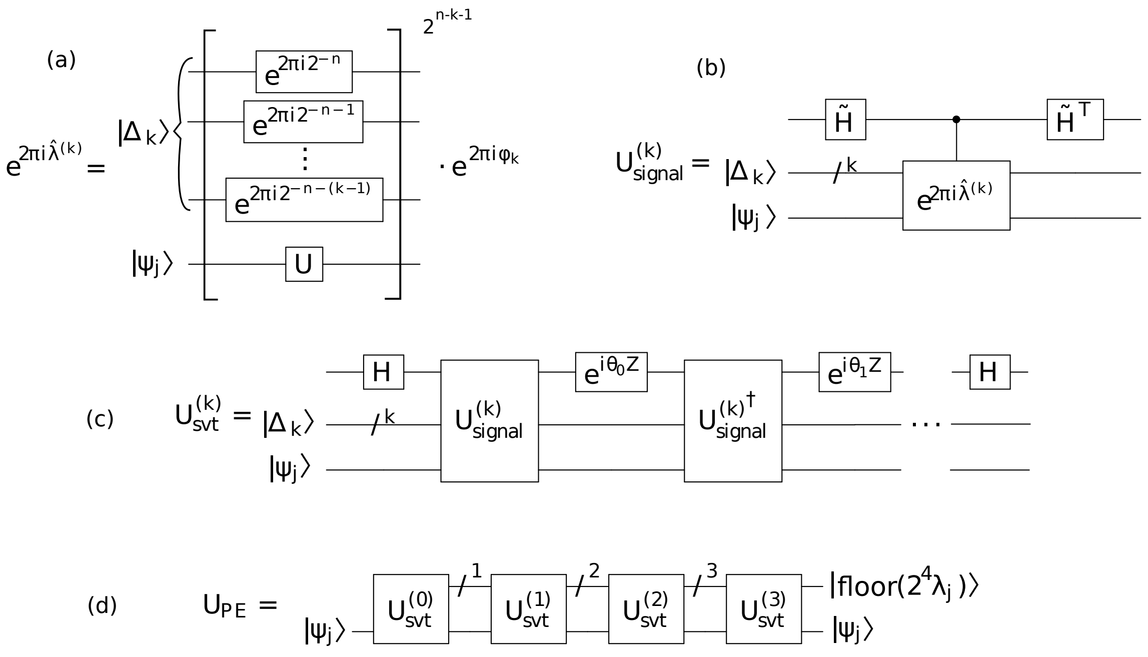


Figure 4: Circuit diagram for the algorithm in Corollary 13. (a) and (b) are depictions of (108) and (113) respectively. (c) depicts the singular value transformation circuit obtained from Lemma 10 which interperses alternating applications of $U^{(k)}_{\text{signal}}$ and $U^{(k)\dagger}_{\text{signal}}$ with phase rotations $e^{i\theta_j Z}$, where the angles θ_j encode the polynomial approximating $A_{\eta\to\delta_{\text{amp}}}(x^2)$. (d) depicts the final circuit assembled via Lemma 7, showing how each iterative estimator's output becomes part of the $|\Delta_k\rangle$ register for the next, and how that register finally becomes the output.

Lemma 14. ***Jacobi-Anger expansion.*** *For any $t \in \mathbb{R}^+$, and any $\varepsilon \in (0, 1/e)$, let:*

$$r(t', \varepsilon') := \text{ the solution to } \varepsilon' = \left(\frac{t'}{r}\right)^r \text{ such that } r \in (t', \infty), \tag{150}$$

$$R := \left\lfloor r\left(\frac{et}{2}, \frac{5}{4}\varepsilon\right)/2 \right\rfloor \tag{151}$$

$$J_k(t) := \text{the } k\text{'th Bessel function of the first kind} \tag{152}$$

$$T_k(t) := \text{the } k\text{'th Chebyshev polynomial of the first kind} \tag{153}$$

$$p_{cos,t}(x) := J_0(t) + 2\sum_{k=1}^{R}(-1)^k J_{2k}(t)T_{2k}(x) \tag{154}$$

Then $p_{cos,t}(x)$ is an even polynomial of degree $2R$ such that for all $x \in [-1, 1]$:

$$|\cos(tx) - p_{cos,t}(x)| \leq \varepsilon. \tag{155}$$

Furthermore:

$$r(t', \varepsilon') \in \Theta\left(t' + \frac{\log \varepsilon'^{-1}}{\log(\log(\varepsilon'^{-1}))}\right) \tag{156}$$

Proof. These results are shown in Lemma 57 and Lemma 59 of [GSLW18], outlined in their section 5.1. □

Again, computation of the degree of the Jacobi-Anger expansion is a bit complicated, but the complexity is worth it due to the method's high performance. Our approach is to first synthesize $\cos\left(\pi\lambda_j^{(k)}\right)$ to obtain a signal that oscillates to indicate $\text{bit}_k(\lambda_j)$, and then apply $A_{\eta\to\delta}(x^2)$ to amplify the signal to 0 or 1, as shown in Figure 2.

Actually, one might observe that synthesizing $\cos\left(\pi\lambda_j^{(k)}\right)$ first is not necessary to make polynomials that look like those in Figure 2. Instead one can take an approach similar to the one used for making rectangle functions in [LC17]: simply shift, scale and add several amplifying polynomials $A_{\eta\to\delta}(x)$ to make the desired shape. None of these operations affect the degree, so this approach also yields the same asymptotic scaling $O(2^n\alpha^{-1}\log(\delta^{-1}))$ as phase estimation. We will see in Corollary 16 that the method using the Jacobi-Anger expansion actually achieves the worse scaling of $O(\alpha^{-1}\log(\delta^{-1})(2^n+\log(\alpha^{-1})))$.

The reason why we present the approach using the Jacobi-Anger expansion, despite it having worse asymptotic scaling, is that in the regime of interest $(n\approx 10, \alpha\approx 2^{-10})$ we numerically find that the Jacobi-Anger expansion actually performs better. There may be a regime where it is better to remove the Jacobi-Anger expansion from the construction, in which case the algorithm is easily adapted.

The rest of the construction of Theorem 15 strongly resembles coherent iterative phase estimation, so much so that we can re-use parts of the proof of Theorem 12. One further difference is that this estimator now has garbage, because we have no guarantee that the block-encoding of the Hamiltonian only has one ancilla.

Theorem 15. ***Coherent Iterative Energy Estimation.*** *Say the block-encoding of H requires a ancillae, that is, U_H acts on $\mathbb{C}^{2^a}\otimes\mathcal{H}$. Then there is an iterative energy estimator with phases and $a+n+3$ qubits of garbage with query complexity:*

$$4\cdot M_{(1-10^{-m_{cos}})\eta_k\to\delta_{amp}}\cdot\left\lfloor r\left(\frac{e}{2}\pi 2^{n-k}, \frac{5}{4}\frac{\eta_k}{2}10^{-m_{cos}}\right)\right\rfloor \tag{157}$$

where $M_{\eta\to\delta}$ and η_k are as in Lemma 11 and δ_{amp} can be chosen to be $(1-10^{-m_{svt}})\delta^2/8$ for any $m_{svt}>0$, and we can choose any $m_{cos}>0$.

Proof. We construct the estimator as follows:

1. Let W_k be a hermitian matrix on k qubits defined by:

$$W_k := 2\sum_{\Delta_k=0}^{2^k-1}\frac{\Delta_k}{2^n}\left|\Delta_k\right\rangle\left\langle\Delta_k\right| \tag{158}$$

W_k has a block-encoding which can be constructed as follows: First, prepare $|+^{n-1}\rangle$:

$$|\Delta_k\rangle\to|\Delta_k\rangle|+^{n-1}\rangle=\frac{1}{\sqrt{2^{n-1}}}\sum_{x=0}^{2^{n-1}-1}|\Delta_k\rangle|x\rangle \tag{159}$$

Next, observe that $\Delta_k\le 2^k-1\le 2^{n-1}-1$. Into an ancilla register compute $|x<\Delta_k\rangle$, and uncompute any garbage necessary to do so.

$$\to\frac{1}{\sqrt{2^{n-1}}}\sum_{x=0}^{2^{n-1}-1}|\Delta_k\rangle|x\rangle|x<\Delta_k\rangle \tag{160}$$

Then postselect that the final register is in the $|1\rangle$ state:

$$\to\frac{1}{\sqrt{2^{n-1}}}|\Delta_k\rangle\sum_{x=0}^{\Delta_k-1}|x\rangle \tag{161}$$

Finally, postselect that the x register is in the $|+^{n-1}\rangle$ state:

$$\to |\Delta_k\rangle \sum_{x=0}^{\Delta_k - 1} \frac{1}{2^{n-1}} = \frac{\Delta_k}{2^{n-1}} |\Delta_k\rangle \tag{162}$$

This process makes use of $n-1$ ancillae initialized and postselected to $|+\rangle$, and one more ancilla that is postselected to $|1\rangle$.

2. Use linear combinations of unitaries to construct a block-encoding of:

$$H^{(k)} := \frac{1}{2} \cdot I \otimes H - \frac{1}{4} \cdot W_k \otimes I + \frac{1}{4} \cdot (4\phi_k 2^{k-n}) \cdot I \otimes I \tag{163}$$

where the ϕ_k are selected just as in as in Theorem 12. Observe that $\phi_k < 1/2$, so therefore $4\phi_k 2^{k-n}$ is a probability which can be block-encoded. Since the block-encoding of H has a ancillae, and that of W_k has n ancillae, and the three terms in the linear combination need two control qubits, the block-encoding of $H^{(k)}$ has $a+n+2$ ancillae.

3. Use Lemma 14 to construct a polynomial approximation $p_{\cos,\pi 2^{n-k}}(x)$ of $\cos(\pi 2^{n-k} x)$ to accuracy $2\delta_{\cos}$, to be picked later.

   Use Lemma 11 to construct a polynomial $A_{(\eta-\delta_{\cos})\to\delta_{\text{amp}}}$. We will pick δ_{amp} later and select η_k just as in Theorem 12.

   Finally, use Lemma 10 to construct U_{svt}, a block-encoding of $\tilde{p}(H^{(k)})$ which approximates the even polynomial:

$$\tilde{p}(x) \approx A_{(\eta_k-\delta_{\cos})\to\delta_{\text{amp}}} \left(p^2_{\cos,\pi 2^{n-k}}(x) \right) \tag{164}$$

   To perform singular value transformation we needed one extra ancilla, so U_{svt} has $a+n+3$ ancillae - these are the garbage output of this map.

4. Use the modified Toffoli gate $I \otimes |0\rangle\langle 0| + X \otimes (I - |0\rangle\langle 0|)$ to conditionally flip the output qubit.

We rewrite $H^{(k)}$ in terms of its eigendecomposition:

$$H^{(k)} = \sum_j \sum_{\Delta_k} \left[\frac{\lambda_j}{2} - \frac{1}{2}\frac{\Delta_k}{2^n} + 2^{k-n}\phi_k \right] |\Delta_k\rangle\langle\Delta_k| \otimes |\psi_j\rangle\langle\psi_j| \tag{165}$$

$$= \sum_j \sum_{\Delta_k} 2^{k-n} \lambda_j^{(k)} |\Delta_k\rangle\langle\Delta_k| \otimes |\psi_j\rangle\langle\psi_j| \tag{166}$$

where we defined $\lambda_j^{(k)} := 2^{n-k-1}\left(\lambda_j - \frac{\Delta_k}{2^n}\right) + \phi_k$ just like in Theorem 12. This protocol implements a map:

$$|0\rangle |0...0\rangle |\Delta_k\rangle |\psi_j\rangle \to \left(\tilde{p}\left(2^{k-n}\lambda_j^k\right) |0\rangle |\text{gar}_{0,j}\rangle + \gamma(\lambda_j^k |1\rangle |\text{gar}_{1,j}\rangle \right) |\Delta_k\rangle |\psi_j\rangle \tag{167}$$

Here $\gamma(\lambda_j^{(k)})$ is defined such that all the other amplitudes for the failed branches of the block-encoding are absorbed into the normalized state $|\text{gar}_{1,j}\rangle$.

We see that $\tilde{p}\left(2^{k-n}\lambda_j^k\right)$ approximates

$$\tilde{p}(\lambda_\Delta/2) \approx A_{(\eta_k-\delta_{\cos})\to\delta_{\text{amp}}} \left(\cos^2\left(\pi \lambda_j^{(k)}\right) \right) \tag{168}$$

which is the same expression encountered in Theorem 12, with the same definition of $\lambda_j^{(k)}$, up to a minor shift on η_k. Therefore, we can follow the same reasoning as in Theorem 12 up to two minor differences, and arrive at the exact same conclusion. Namely, if we select some $m_{\text{svt}} > 0$ and then let:

$$\delta_{\text{amp}} := (1 - 10^{-m_{\text{svt}}})\frac{\delta^2}{8}, \qquad \delta_{\text{svt}} := 10^{-m_{\text{svt}}}\frac{\delta^2}{8}, \tag{169}$$

Then the unitary channel we implement is at most δ-far in diamond norm to a channel that implements the map:

$$|0\rangle\,|0...0\rangle\,|\Delta_k\rangle\,|\psi_j\rangle \to e^{i\varphi_j}\,|\text{bit}_k(\lambda_j)\rangle\,|\text{gar}_{\lambda_j}\rangle\,|\Delta_k\rangle\,|\psi_j\rangle \tag{170}$$

whenever Δ_k encodes the last k bits of λ_j.

Since the argument in Theorem 12 is lengthy and the modifications are extremely minor, we will not repeat the argument here. Instead we will articulate the two things that change.

First, Theorem 12 gives an estimator with phases and no garbage, whereas here we also have garbage. The garbage just tags along for the entire calculation, and when we come to selecting φ_j depending on λ_j we can also select $|\text{gar}_{\lambda_j}\rangle = |\text{gar}_{0/1,j}\rangle$ depending on $\text{bit}_k(\lambda_j)$.

Second, we incur an error of $\delta_{\cos}$ in the approximation of $\cos(\pi 2^{n-k}x)$ with $p_{\cos,\pi 2^{n-k}}(x)$. Since we show that $\cos^2(\pi 2^{n-k}x)$ is bounded away from $\frac{1}{2}$ by $\pm\eta$ we therefore have that $p_{\cos,\pi 2^{n-k}}(x)^2$ is bounded away from $\frac{1}{2}$ by $\pm\left(\eta - \frac{2\delta_{\cos}}{2}\right)$. The amplifying polynomial then proceeds to amplify $(\eta - \delta_{\cos}) \to \delta_{\text{amp}}$ appropriately, so the calculation proceeds the same. We just need to ensure that $\eta - \delta_{\cos} > 0$, so we select

$$\delta_{\cos} := 10^{-m_{\cos}} \cdot \eta \tag{171}$$

for some $m_{\cos} > 0$. This completes the accuracy analysis.

Finally, we analyze the query complexity. The block-encoding of $H^{(k)}$ makes one query to U_H, so by Lemma 10 the query complexity is exactly the degree of $\tilde{p}(x)$ which is the degree of $A_{(\eta-\delta_{\cos})\to\delta_{\text{amp}}}\left(p^2_{\cos,\pi 2^{n-k}}(x)\right)$. From Lemma 14 and Lemma 11, the degree is:

$$M_{(\eta-\delta_{\cos})\to\delta_{\text{amp}}} \cdot 2 \cdot 2\left\lfloor r\left(\frac{e}{2}\pi 2^{n-k}, \frac{5}{4}\frac{\delta_{\cos}}{2}\right)\right\rfloor \tag{172}$$

Substituting the definitions for $\eta, \delta_{\text{amp}}$ and $\delta_{\cos}$ yields the final runtime. As with Theorem 12, δ_{amp} can be made larger by increasing m_{amp}. By increasing $m_{\cos}$ we can decrease $\delta_{\cos}$, which decreases the degree of $A_{(\eta-\delta_{\cos})\to\delta_{\text{amp}}}(x)$ while increasing the degree of $p_{\cos,\pi 2^{n-k}}(x)$. The Jacobi-Anger expansion deals with error more efficiently than the amplifying polynomial, so in practice $m_{\cos}$ should be quite large. □

As with phase estimation, we also pack the coherent iterative energy estimator into a regular energy estimator using Lemma 7. This time, since the iterative estimator produces garbage it makes sense to uncompute the garbage using Lemma 8.

Corollary 16. ***Improved Energy Estimation.*** *The iterative energy estimator with phases and garbage from Theorem 15 can be combined with Lemma 8 and Lemma 7 to make an energy estimator without phases and without garbage with query complexity bounded by:*

$$O\left(\alpha^{-1}\log(\delta^{-1})\left(2^n + \log\left(\alpha^{-1}\right)\right)\right) \tag{173}$$

assuming that α is bounded away from 1 by a constant.

Furthermore, even when there is no rounding promise, there exists an algorithm that, given an eigenstate $|\psi_j\rangle$ of the Hamiltonian λ_j, for any $\delta > 0$ performs a transformation δ-close in diamond norm to the map:

$$|\psi_j\rangle\langle\psi_j| \to \left(p\,|\text{floor}(2^n\lambda_j)\rangle\langle\text{floor}(2^n\lambda_j)| + (1-p)\,|\lambda_j'\rangle\langle\lambda_j'|\right)|\psi_j\rangle\langle\psi_j| \tag{174}$$

where p is some probability and $\lambda_j' = \text{floor}(2^n\lambda_j) - 1 \bmod 2^n$ is an erroneous estimate. Just as in Corollary 13, the performance is the same except that $0 < \alpha < 1$ can be any constant.

Proof. As with Corollary 13, we write η_k to make the k dependence explicit and demand accuracy $\delta_{\text{amp},k} = (1 - 10^{-m_{\text{amp}}})(\delta 2^{-k-1})^2/8$ for the k'th bit. From Lemma 14 we obtain an asymptotic upper bound $r(t,\varepsilon) \in O\left(t + \log(\varepsilon^{-1})\right)$. Again, recall from Lemma 11 that $M_{\eta_k \to \delta_{\text{amp}}} \in O\left(\eta_k^{-1}\log(\delta_{\text{amp}}^{-1})\right)$. The asymptotic query complexity of the iterative energy estimator from Theorem 15 is then bounded by:

$$O\left((\eta_k - \delta_{\cos})^{-1}\log(\delta_{\text{amp},k}^{-1})\cdot(2^{n-k} + \log(\delta_{\cos,k}))\right) \tag{175}$$

$$\leq O\left((1-10^{-m_{\cos}})^{-1}\eta_k^{-1}\log((1-10^{-m_{\text{amp}}})^{-1}2^{2(k+1)}\delta^{-2}8)\cdot(2^{n-k} + \log(10^{m_{\cos}}\eta_k^{-1}))\right) \tag{176}$$

$$\leq O\left(\eta_k^{-1}\log(2^{k+1}\delta^{-1})\cdot(2^{n-k} + \log(\eta_k^{-1}))\right) \tag{177}$$

Next we invoke Lemma 8 to remove the phases and the garbage, doubling the query complexity. We do this before invoking Lemma 7, because Lemma 7 involves blowing up the number of garbage registers by a factor of n. While we could also invoke Lemma 7 and then invoke Lemma 3 to obtain a map without garbage, this would involve many garbage registers sitting around waiting to be uncomputed for a long time. If we invoke Lemma 8 first we get rid of the garbage immediately.

Finally, we invoke Lemma 7 to turn our iterative energy estimator without garbage and phases into a regular energy estimator without garbage and phases. As in Corollary 13, we observe that $\eta_0 = \alpha/2$ and for $k > 0$ we have η_k bounded from below by a constant. Then, the total query complexity is:

$$O\left(\eta_0^{-1}\log(2^{0+1}\delta^{-1})\left(2^{n-0} + \log\left(\eta_0^{-1}\right)\right) + \sum_{k=1}^{n-1}\eta_k^{-1}\log(2^{k+1}\delta^{-1})\left(2^{n-k} + \log\left(\eta_k^{-1}\right)\right)\right) \tag{178}$$

$$\leq O\left(\alpha^{-1}\log(\delta^{-1})\left(2^n + \log\left(\alpha^{-1}\right)\right) + \sum_{k=1}^{n-1}(k + \log(\delta^{-1}))2^{n-k}\right) \tag{179}$$

$$\leq O\left(\alpha^{-1}\log(\delta^{-1})\left(2^n + \log\left(\alpha^{-1}\right)\right) + 2(2^n - n - 1) + (2^n - 2)\log(\delta^{-1})\right) \tag{180}$$

$$\leq O\left(\alpha^{-1}\log(\delta^{-1})\left(2^n + \log\left(\alpha^{-1}\right)\right)\right) \tag{181}$$

Next, we show that it is possible to implement a map that, given an eigenstate $|\phi_j\rangle$, measures an estimate that is either $\text{floor}(2^n\lambda_j)$ or $\text{floor}(2^n\lambda_j) - 1 \bmod 2^n$ with some probability. Note that this is not the same algorithm as above. Just as with phase estimation, it is only the first bit that actually needs the rounding promise, and all other bits are guaranteed to be deterministic.

The first bit performs a map of the form:

$$|0^n\rangle\,|0...0\rangle\,|\psi_j\rangle \to \left(\sqrt{p}\,|0\rangle\,|\text{gar}_{0,j}\rangle + \sqrt{1-p}\,|1\rangle\,|\text{gar}_{1,j}\rangle\right)|\psi_j\rangle \tag{182}$$

We immediately see that Lemma 8 cannot be used to perform uncomputation here, because uncomputation only works when $p = 1$ or $p = 0$. Instead, we simply measure the output register containing $|0\rangle$ or $|1\rangle$ and discard the garbage. This would damage any superposition over the $|\psi_j\rangle$, which is why this algorithm only works if the input is an eigenstate.

We then use iterative estimators for the remaining bits to compute the rest of the estimate. This time their outputs will be deterministic, but there is no point in doing uncomputation since the superposition has already collapsed. Instead, we do the same thing as for the $k = 0$ estimator: compute the next bit and some garbage, and discard the garbage. The final answer is either $\text{floor}(2^n\lambda_j)$ or $\text{floor}(2^n\lambda_j) - 1$ for the same reason as in Corollary 13.

□

3 Performance Comparison

Above, we have presented a modular framework for phase and energy estimation using the key ingredients in Theorem 12 and Theorem 15 respectively. These results already demonstrate several advantages over textbook phase estimation as presented in Proposition 5.

First, they eliminate the QFT and do not require a sorting network to perform median amplification. Instead, they rely on just a single tool: singular value transformation.

Second, they require far fewer ancillae. Our improved phase estimation algorithm requires no ancillae at all, and is merely 'with phases' so arguably does not even need uncomputation for some applications. On the other hand, textbook phase estimation requires $O((n + \log(\alpha^{-1}))\log(\delta^{-1}))$ ancillae in order to implement median amplification.

Given a block-encoding of a Hamiltonian with a ancillae, then our energy estimation algorithm requires $a + n + 3$ ancillae. But in order to even compare Proposition 5 to Theorem 15 we need a method to perform energy estimation using textbook phase estimation. This is achieved through Hamiltonian simulation.

Which method of Hamiltonian simulation is best depends on the particular physical system involved. Hamiltonian simulation using the Trotter approximation can perform exceedingly well in many situations [SHC20]. However, in our analysis we must be agnostic to the particular Hamiltonian in question, and furthermore need a unified method for comparing the performance. Hamiltonian simulation via singular value transformation [LC17, GSLW18], lets us compare Proposition 5 and Theorem 15 on the same footing. After all, this method features the best known asymptotic performance in terms of the simulation time [Childs&20] in a black-box setting.

Singular value transformation constructs an approximate block-encoding of e^{iHt} with ancillae. Since e^{iHt} is unitary, in the ideal case the ancillae start in the $|0\rangle$ state and are also guaranteed to be mapped back to the $|0\rangle$ state. But in the approximate case the ancillae are still a little entangled with the remaining registers, so they become an additional source of error. Certainly the ancillae cannot be re-used to perform singular value transformation again, because then the errors pile up with each use. Thus, we are in a similar situation to Lemma 3, where we must discard some qubits and take into account the error.

Therefore, we must do some additional work beyond the Hamiltonian simulation method presented in [GSLW18], to turn the approximate block-encoding of a unitary into a channel that approximates the unitary in diamond norm. The main trick for this proof is to consider postselection of the ancilla qubits onto the $|0\rangle$ state. Then the error splits into two parts: the error of the channel when the postselection succeeds, and the probability that the postselection fails.

The Hamiltonian simulation method is extremely accurate, letting us obtain block-encodings with an error that decays exponentially in the query complexity to U_H. Thus, the error contribution of this channel is almost entirely negligible in the performance analysis. The purpose of the argument below is really to provide a Lemma analogous to Lemma 4 that lets us convert error bounds in spectral norms on block-encodings to diamond norms. That way, our entire error analysis is completely formal, as it should be for a fair comparison of all algorithms involved. Notably, an ε-accurate block-encoding of a unitary is not the same thing as an ε-accurate implementation of that unitary: the latter implies a channel with diamond norm error 2ε while the prior yields an error 4ε due to the ancilla registers.

Lemma 17. ***Hamiltonian simulation.*** *Say U_H is a block-encoding of a Hamiltonian H. Then, for any $t > 0$ and $\varepsilon > 0$ there exists a quantum channel that is ε-close in diamond norm to the channel induced by the unitary e^{iHt}. This channel can be implemented using*

$$3 \cdot r\left(\frac{et}{2}, \frac{\varepsilon}{24}\right) + 3 \tag{183}$$

queries to controlled-U_H or controlled-$U_H^\dagger$.

Proof. This is an extension of Theorem 58 of [GSLW18], which states that there exists a block-encoding U_A of a matrix A such that $|A - e^{iHt}| \leq \varepsilon$ with query complexity $3r\left(\frac{et}{2}, \frac{\varepsilon}{6}\right)$. This result leverages the Jacobi-Anger expansion (Lemma 14) to construct approximate block-encodings of $\sin(tH)$ and $\cos(tH)$ and uses linear combinations of unitaries to approximate $e^{iHt}/2$. Then it uses oblivious amplitude amplification to get rid of the factor of $1/2$, obtaining U_A. If U_H is a block-encoding with a ancillae, then U_A has $a+2$ ancillae.

All that is left to do is to turn this block-encoding into a quantum channel that approximately implements e^{iHt}. To do so, we just initialize the ancillae to $|0^{a+2}\rangle$, apply U_A, and trace out the ancillae. We can write this channel Λ as:

$$\Lambda(\rho) := \sum_i (\langle i| \otimes I) U_A (|0^{a+2}\rangle \langle 0^{a+2}| \otimes \rho) U_A^\dagger (|i\rangle \otimes I) \tag{184}$$

To finish the theorem we must select an ε so that the error in diamond norm of Λ to the channel implemented by e^{iHt} is bounded by δ. To do so, we write Λ as a sum of postselective channels $\Lambda_i(\rho)$:

$$\Lambda_i(\rho) := (\langle i| \otimes I) U_A (|0\rangle \langle 0| \otimes \rho) U_A^\dagger (|i\rangle \otimes I) \tag{185}$$

That way $\Lambda = \sum_i \Lambda_i$. If we let $\Gamma_{e^{iHt}} := e^{iHt} \rho e^{-iHt}$, then we can bound the error in diamond norm using the triangle inequality:

$$|\Lambda - \Gamma_{e^{iHt}}|_\diamond \leq |\Lambda_0 - \Gamma_{e^{iHt}}|_\diamond + \left|\sum_{i>0} \Lambda_i\right|_\diamond \tag{186}$$

Now we proceed to bound the two terms individually. Observe that:

$$\Lambda_0(\rho) := (\langle 0| \otimes I) U_A (|0\rangle \langle 0| \otimes \rho) U_A^\dagger (|0\rangle \otimes I) = A \rho A^\dagger \tag{187}$$

Since $|A - e^{iHt}| \leq \varepsilon$, we can invoke Lemma 4 and see that $|\Lambda_0 - \Gamma_{e^{iHt}}|_\diamond \leq 2\varepsilon$, and we are done with the first term.

To bound $|\sum_{i>0} \Lambda_i|_\diamond$, we first observe that $\sum_{i>0} \Lambda_i = \Lambda - \Lambda_0$. Second, we observe that the Λ_i are all positive semi-definite, so $|\Lambda_i(\rho)|_1 = \mathrm{Tr}(\Lambda_i(\rho))$. Plugging in the definition of the diamond norm, we can compute:

$$\left|\sum_{i>0} \Lambda_i\right|_\diamond = \sup_\rho \left|\sum_i (\Lambda_i \otimes \mathcal{I})(\rho)\right|_1 \tag{188}$$

$$\leq \sup_\rho \mathrm{Tr}\left(\sum_i (\Lambda_i \otimes \mathcal{I})(\rho)\right) \tag{189}$$

$$= \sup_\rho \mathrm{Tr}\left((\Lambda \otimes \mathcal{I})(\rho) - (\Lambda_0 \otimes \mathcal{I})(\rho)\right) \tag{190}$$

$$= 1 - \inf_\rho \mathrm{Tr}(A\rho A^\dagger) \tag{191}$$

If we let $E := e^{iHt} - A$ so that $|E| \leq \varepsilon$, and plug into the above expression, we get:

$$\mathrm{Tr}(A\rho A^\dagger) = \mathrm{Tr}(e^{iHt}\rho e^{-iHt} - E\rho e^{-iHt} - e^{iHt}\rho E^\dagger + E\rho E^\dagger) \geq 1 - 2\varepsilon + \varepsilon^2 \tag{192}$$

Putting everything together we obtain $|\Lambda - \Gamma_{e^{iHt}}|_\diamond \leq 2\varepsilon + 2\varepsilon - \varepsilon^2 \leq 4\varepsilon$. So if we select $\varepsilon := \delta/4$ we obtain the desired bound.

□

The above proof uses the trick for the proof of the block-measurement theorem that we mentioned in the introduction. We will need it again in the proof Theorem 19. Thus, while we are at it, we may as well state the generalization of this result to any block-encoded unitary as a proposition.

Proposition 18. *Say U_A is a block-encoding of A, which is ε-close in spectral norm to a unitary V. Then there exists a quantum channel 4ε-close in diamond norm to the channel $\rho \to V\rho V^\dagger$.*

Proof. Lemma 17 proved this result with $V = e^{iHt}$. The exact same argument holds for abstract V. □

Returning to the ancilla discussion, say that the block-encoding of the input Hamiltonian has a ancillae. Then we see that Theorem 15 requires $a+n+3$ ancillae, while textbook phase estimation combined with Lemma 17 requires $O(a + (n + \log(\alpha^{-1}))\log(\delta^{-1}))$. This is because the extra ancillae required to perform the procedure in Lemma 17 can be discarded and reset after each application of e^{iHt}. Also note that despite the fact that the above implementation of e^{iHt} is not unitary, the overall protocol in Proposition 5 is still approximately invertible as required by Lemma 3, since we can just use Lemma 17 to implement e^{-iHt} instead.

Next, we compare the algorithms in terms of their query complexity to the unitary U or the block-encoding U_H. Note that this is *not* the gate complexity: the number of gates from some universal gate set used to implement the algorithm. The reason for this choice is that the gate complexity of U or U_H depends on the application, and is likely much much larger than any of the additional gates used to implement singular value transformation. The algorithms' query complexities depend on three parameters: α, n, and δ. In the following we discuss the impact of these parameters on the complexity and show how we arrive at the 14x and 20x speedups stated in the introduction.

The performance in terms of α is plotted in Figure 5, for fixed $n = 10$ and $\delta = 10^{-30}$. This comparison shows several features. First, we see that the novel algorithms in this paper are consistently faster than the traditional methods in this regime. The only exception is Corollary 13 combined with Lemma 3 to remove the phases, which is outperformed by traditional phase estimation for some value of $\alpha > 1/2$. Such enormous α is obviously impractical: even traditional phase estimation does not round to the nearest bit in this regime.

Second, the performance of Proposition 5 exhibits a ziz-zag behavior. This is because we reduce α by estimating more bits than we need and then rounding them away, a process that can only obtain α of the form 2^{-1-r} for integer r. When $\alpha > \frac{1}{2}$ then we no longer use the rounding strategy since we can achieve these values with median amplification alone. Therefore, for large enough α the line is no longer a zig-zag and begins to be a curve with shape $\sim \alpha^{-2}$.

The zig-zag behavior makes comparison for continuous values of α complicated. In our analysis we would like to compare against the most efficient version of traditional phase estimation, so we continue our performance analysis where α is a power of two, maximizing the efficiency (but minimizing our speedup). In Figure 6 we show the performance when vary α, n and δ independently.

We see that once $n \gtrsim 10$, $\alpha \lesssim 2^{-10}$ and $\delta \lesssim 10^{-30}$ the speedup stabilizes at about 14x for phase estimation and 20x for energy estimation. Our method therefore shows an significant improvement over the state of the art.

Of course, several assumptions needed to be made in order to claim a particular multiplicative speedup. Many of these assumptions were made specifically to maximize the performance of the traditional method. For example, we count performance in terms of query complexity rather than gate complexity, which neglects all the additional processing that phase estimation needs to perform for median amplification. This method of comparison favors the traditional method, since it neglects the fact that our new methods require significantly less ancillary processing. Furthermore, we select α to be a power of two, so that phase estimation is maximally efficient. However, we also assume that n is large enough and that α and δ are small enough that the speedup is stable. If the accuracy required is not so large, then the speedup is less significant.

Which method is best in reality will depend on the situation. In particular, the appropriate choice of α when studying real-world Hamiltonians remains an interesting direction of study. In reality it will not be possible to guarantee a rounding promise, so one's only option is to pick a small value of α and hope for the best. How small of an α is required?

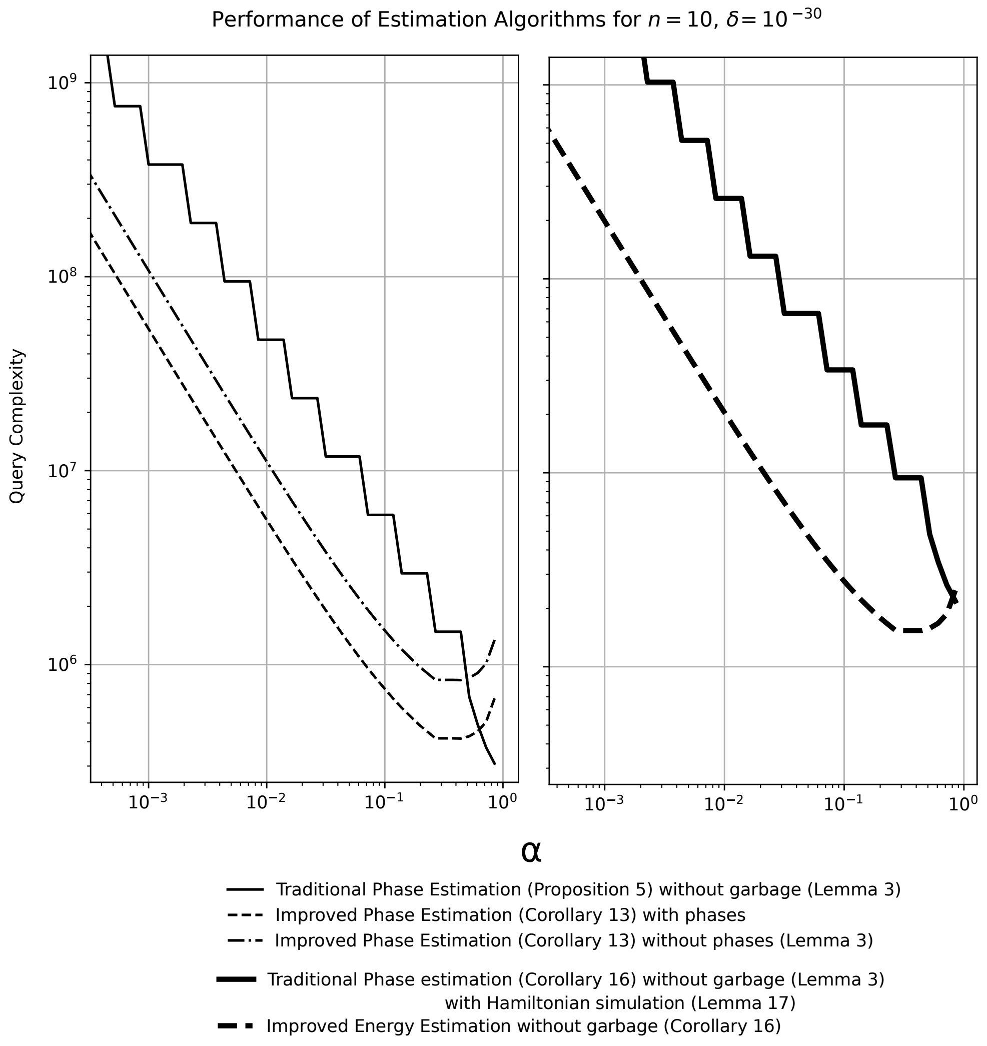


Figure 5: Performance of estimation algorithms presented in this paper. Phase estimation algorithms are presented with slim lines on the left, and energy estimation algorithms are presented with thick lines on the right. The zig-zag behavior of phase estimation is explained by the method by which Proposition 5 reduces α: by estimating more bits than needed and then rounding them. Thus, the α achieved by phase estimation is always a power of two.

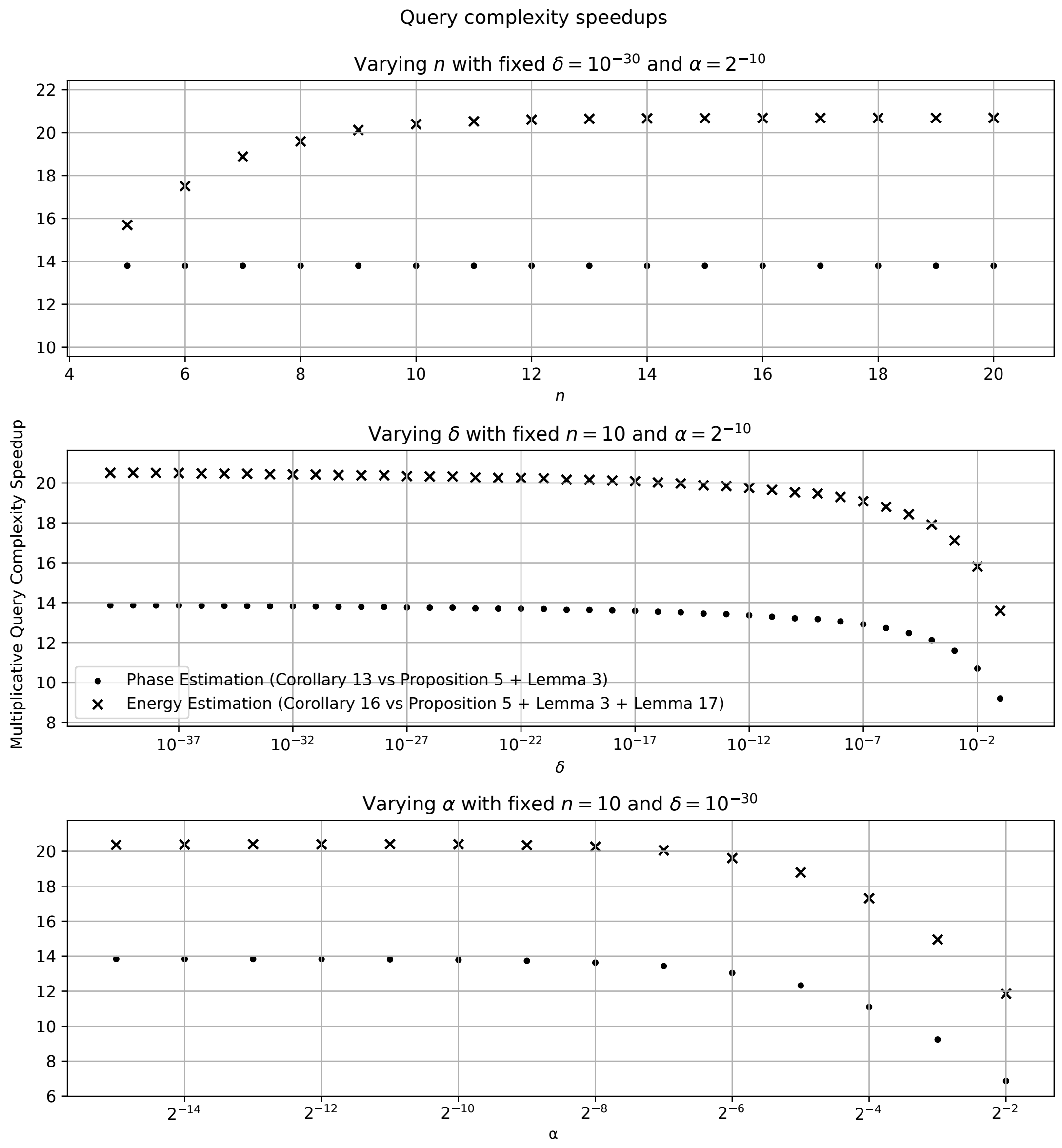


Figure 6: Speedup over traditional methods by our new methods for phase and energy estimation. When $n \gtrsim 10$, $\delta \lesssim 10^{-30}$ and $\alpha \lesssim 2^{-10}$ the speedups become stable. Together with Figure 5 we can conclude that the speedup for phase estimation is about 14x and the speedup for energy estimation is about 20x.

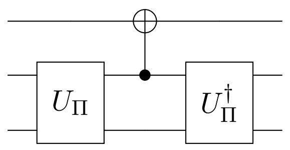
4 Block-Measurement

We have presented improved algorithms for phase and energy estimation. In this section we prove the block-measurement theorem from the introduction. The goal of the block-measurement protocol is the following: given an approximate block-encoding of a projector Π, implement a channel close to the unitary:

$$|1\rangle \otimes \Pi + |0\rangle \otimes (I - \Pi) \tag{193}$$

When the block-encoding of Π is exact, then the above is easily accomplished through linear combinations of unitaries and oblivious amplitude amplification. In particular, we can rewrite the above as $|0\rangle \otimes I - \sqrt{2}\,|-\rangle \otimes \Pi$, so we need to amplify away a factor of $1/(1+\sqrt{2})$ from the linear combination. Following Theorem 28 of [GSLW18], we observe that $T_5(x)$ is the first Chebyshev polynomial that has a solution $T_5(x) = \pm 1$ such that $x < 1/(1+\sqrt{2})$. We nudge the factor down to the solution x with some extra postselection, and then we meet the conditions of this theorem. This demands five queries to the block-encoding of Π. Then we invoke our Proposition 18 to turn the block-encoding of $|1\rangle \otimes \Pi + |0\rangle \otimes (I - \Pi)$ into a channel.

However, the above strategy is both more complicated and more expensive than necessary - we can do this in just two queries. Say U_Π is a block-encoding of Π with m ancillae. Then, let V_Π:

$V_\Pi :=$  (194)

where the CNOT above refers to $X \otimes |0^m\rangle\langle 0^m| + I \otimes (I - |0^m\rangle\langle 0^m|)$. Then, V_Π satisfies:

$$(I \otimes \langle 0^m| \otimes I) V_\Pi (|0\rangle \otimes |0^m\rangle \otimes I) \tag{195}$$
$$= |1\rangle \otimes \Pi + |0\rangle \otimes (I - \Pi) \tag{196}$$

So viewing the $|0^m\rangle$ register as the postselective part of a block-encoding, we see that V_Π is a block-encoding of the desired operation. Then we invoke Proposition 18 to construct a channel Λ_Π from V_Π. Now all that is left to do is the error analysis.

In fact, when $m = 1$, then there is an extent that we do not even need the modified CNOT gate and can get away with just a single query. We used this fact in Theorem 12. This is because if $\Pi = \sum_j \alpha_j |\psi_j\rangle\langle\psi_j|$, then we can write:

$$U_\Pi(|0\rangle \otimes I) = \sum_j (\alpha_j |0\rangle + \beta_j |1\rangle) |\psi_j\rangle\langle\psi_j| \tag{197}$$

where β_j is some amplitude satisfying $|\beta_j|^2 + |\alpha_j|^2 = 1$. To compare, the desired operation is:

$$\sum_j (\alpha_j |1\rangle + (1-\alpha_j)|0\rangle) |\psi_j\rangle\langle\psi_j| \tag{198}$$

Since $\alpha_j \in \{0, 1\}$, we know that $\beta_j = e^{i\phi_j}(1 - \alpha_j)$ for some phase ϕ_j that may depend on $|\psi_j\rangle$. So, with the exception of the phase correction, we are already one X gate away from the desired unitary.

Is it possible to remove this phase correction without uncomputation? If U_Π is obtained through singular value transformation of some real eigenvalues λ_j then we actually have more control than Lemma 10 would indicate. Looking at Theorem 3 of [GSLW18], we can actually select polynomials P, Q such that $\alpha_j = P(\lambda_j)$ and $\beta_j = Q(\lambda_j)$, provided that P, Q satisfy some conditions including $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1$. For what positive-real-valued polynomials $P(x)$ is it possible to choose a positive-real-valued $Q(x)$ that satisfies this relation, thus removing the phases $e^{i\phi_j}$? We leave this question for future work.

Now we proceed to the formal statement and error analysis of the block-measurement theorem.

Theorem 19. ***Block-measurement.*** *Say Π is a projector, and A is a hermitian matrix satisfying $|\Pi - A^2| \leq \varepsilon$. Also, A has a block-encoding U_A. Then the channel Λ_A, constructed in the same way as Λ_Π above just with U_A instead of U_Π, approximates Λ_Π in diamond norm:*

$$|\Lambda_A - \Lambda_\Pi|_\diamond \leq 4\sqrt{2}\varepsilon \tag{199}$$

Proof. Just like V_Π, V_A is a block-encoding of the unitary:

$$X \otimes A^2 + I \otimes (I - A^2) \tag{200}$$

The distance in spectral norm to $V_\Pi = X \otimes \Pi + I \otimes (I - \Pi)$ is:

$$\begin{aligned}
&\Big| |1\rangle \otimes \Pi + |0\rangle \otimes (I-\Pi) - |1\rangle \otimes A^2 - |0\rangle \otimes (I - A^2)\Big| && (201)\\
=&\Big| |1\rangle \otimes (\Pi - A^2) + |0\rangle \otimes (A^2 - \Pi)\Big| && (202)\\
=&\sqrt{2}\left|\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes (\Pi - A^2)\right| && (203)\\
=&\sqrt{2}\left|\Pi - A^2\right| \leq \sqrt{2}\varepsilon && (204)
\end{aligned}$$

So we have a block-encoding of a matrix $\sqrt{2}\varepsilon$-close in spectral norm to the unitary $X \otimes \Pi + I \otimes (I - \Pi)$. Now we just use Proposition 18, which gives a channel with diamond norm accuracy $4\sqrt{2}\varepsilon$. Then we get the desired operation by initializing an extra qubit to $|0\rangle$ before applying the channel. □

What happens when A is not a projector? For simplicity, say the input state to the circuit is $|\Psi\rangle$, which happens to be an eigenstate of A. Then:

$$V_A\left(|0\rangle \otimes |0^m\rangle \otimes |\Psi\rangle\right) = \left(|1\rangle \otimes U_A^\dagger |0^m\rangle\langle 0^m| U_A |0^m\rangle + |0\rangle \otimes U_A^\dagger\left(I - |0^m\rangle\langle 0^m|\right) U_A |0^m\rangle\right) \otimes |\Psi\rangle \tag{205}$$

$$= \left(-\sqrt{2}\,|-\rangle \otimes U_A^\dagger |0^m\rangle\langle 0^m| U_A |0^m\rangle + |0\rangle \otimes U_A^\dagger U_A |0^m\rangle\right) \otimes |\Psi\rangle \tag{206}$$

We are interested in the state obtained by tracing out the $|0^m\rangle$ register above. For $j \in \{0,1\}^m$, we let γ_j abbreviate a matrix element of U_A:

$$\gamma_j |\Psi\rangle := (\langle 0^m| \otimes I) U_A (|j\rangle \otimes |\Psi\rangle) \tag{207}$$

$$(I \otimes \langle j| \otimes I)\, V_A\left(|0\rangle \otimes |0^m\rangle \otimes |\Psi\rangle\right) = \left(-\gamma_j^\dagger \gamma_0 \sqrt{2}\,|-\rangle + \delta_{j,0}\,|0\rangle\right) \otimes |\Psi\rangle \tag{208}$$

Note that γ_0 is the eigenvalue of A corresponding to $|\Psi\rangle$. Then, the reduced density matrix after tracing out the middle register is:

$$\sum_{j\in\{0,1\}^m} \Big(-\gamma_j^\dagger\gamma_0\sqrt{2}\,|-\rangle + \delta_{j,0}\,|0\rangle\Big)\Big(-\gamma_j\gamma_0^\dagger\sqrt{2}\,\langle-| + \delta_{j,0}\,\langle 0|\Big) \otimes |\Psi\rangle\langle\Psi| \tag{209}$$

$$= \sum_{j\in\{0,1\}^m} \Big(2|\gamma_j|^2|\gamma_0|^2\,|-\rangle\langle-| - \sqrt{2}\delta_{j,0}\gamma_j\gamma_0^\dagger\,|0\rangle\langle-| - \sqrt{2}\delta_{j,0}\gamma_j^\dagger\gamma_0\,|-\rangle\langle 0| + \delta_{j,0}\,|0\rangle\langle 0|\Big) \otimes |\Psi\rangle\langle\Psi| \tag{210}$$

$$= \Big(2|\gamma_0|^2\,|-\rangle\langle-| - \sqrt{2}|\gamma_0|^2\,|0\rangle\langle-| - \sqrt{2}|\gamma_0|^2\,|-\rangle\langle 0| + |0\rangle\langle 0|\Big) \otimes |\Psi\rangle\langle\Psi| \tag{211}$$

$$= \Big(|\gamma_0|^2\,|1\rangle\langle 1| + (1-|\gamma_0|^2)\,|0\rangle\langle 0|\Big) \otimes |\Psi\rangle\langle\Psi| \tag{212}$$

This somewhat complicated calculation produced a simple result: tracing out the $|0^m\rangle$ register collapses the superposition on the output register. A more general version of this calculation where the input state is not an eigenstate of A demonstrates that this collapse also damages the superposition on the input register (although it does not fully collapse it). Intuitively, we can see why this is the case even without doing the full calculation: since measuring the middle register collapses the superposition that encodes the eigenvalues, the environment that traced out the middle register thus learns information about the eigenvalues. But the distribution over eigenvalues also contains information about the input superposition over eigenvectors, so therefore the input state will be damaged.

In essence, we have re-derived the fact that uncomputation is impossible unless the output of the computation is deterministic.

5 Non-Destructive Amplitude Estimation

In this section we show how to use our energy estimation algorithm to perform amplitude estimation. The algorithms for energy and phase estimation demanded a rounding promise on the input Hamiltonian, which guarantees that they do not damage the input state even if it is not an eigenstate of the unitary or Hamiltonian of interest. However, as we shall see, this is not a concern for amplitude estimation. This is because we can construct a Hamiltonian whose eigenstate is the input state.

Say Π is some projector and $|\Psi\rangle$ is a quantum state. Non-destructive amplitude estimation obtains an estimate of $a = |\Pi\,|\Psi\rangle\,|$ given exactly one copy of $|\Psi\rangle$, and leaves that copy of $|\Psi\rangle$ intact. This subroutine is explicitly required by the algorithms in [HW19, AHNTW20], which can be used to perform Bayesian inference, thermal state preparation and partition function estimation. However, it is also quite useful in general when $|\Psi\rangle$ is expensive to prepare. For example, when estimating the expectation of an observable on the ground state of a Hamiltonian, preparing the ground state can be very expensive. Thus, it may be practical to only prepare it once.

The only previously known algorithm for non-destructive amplitude estimation is given in [HW19]. It works via several invocations of amplitude estimation according to [BHMT00], which is based on phase estimation. We argue that our new algorithm has several performance advantages over the prior art. However, these advantages are not very quantifiable, preventing us from computing a numerical constant-factor speedup. The advantages are as follows:

- Our new algorithm requires dramatically fewer ancillae. This is because [HW19] relies on phase estimation with median amplification. As argued above, median amplification requires $O(n\log(\delta^{-1}))$ ancillae. In contrast, we simply use the protocol for energy estimation which requires $n + O(1)$ ancillae.

- Our new algorithm runs in a fixed amount of time: one application of our energy estimation algorithm suffices. In contrast, [HW19]'s algorithm works by repeatedly attempting to 'repair' the input state. This process succeeds only with probability 1/2, so while the expected number of attempts is constant, the resulting algorithm is highly adaptive and may need a variable amount of time.

- Our new algorithm does not require knowledge of a lower bound on the amplitude a. The 'repair' step in [HW19] itself involves another application of phase estimation, which must produce an estimate with enough accuracy to distinguish $\arcsin(a)$ and $-\arcsin(a)$. This also implies that [HW19] can only produce relative-error estimates, even when an additive-error estimate might be sufficient, as it is in [AHNTW20].

- While, due to the above differences, it is not really possible to perform a side-by-side constant-factor comparison of the algorithms, we do expect to inherit a modest constant-factor speedup from our energy estimation algorithm. [BHMT00] has no need to perform Hamiltonian simulation, so we expect the speedup to look more like the one in the case of phase estimation. Since no rounding promise is required, making α tiny is not really necessary. But it *is* necessary that $\alpha < 1/2$ since this makes traditional phase estimation round correctly. Looking at Figure 5, we already see constant factor speedups in this regime.

Another minor advantage of our method over [HW19] is that it estimates a^2 directly, rather than going through the Grover angle $\theta := \arcsin(a)$. Coherently evaluating a sine function in superposition to correct this issue, while possible, will have an enormous ancilla overhead. a^2 is the probability with which a measurement of $|\Psi\rangle$ yields a state in Π, which may be useful directly.

We note that an additive error estimate of a is slightly stronger than an additive error estimate of a^2. Moreover this accuracy seems to be necessary for some versions of quantum mean estimation: see for example Appendix C of [AHNTW20]. If the amplitude a is desired rather than a^2, then, since the algorithm proceeds by obtaining a block-encoding of a^2, one can use singular value transformation to make a block-encoding of a instead. A polynomial approximation of $\sqrt{x}$ could be constructed from Corollary 66 of [GSLW18]. We leave a careful error analysis of a direct estimate of a to future work.

Corollary 20. ***Non-destructive amplitude estimation.*** *Say Π is a projector and R_Π is a unitary that reflects about this projector:*

$$R_\Pi := 2\Pi - I \tag{213}$$

Let $|\Psi\rangle$ be some quantum state such that $a := |\Pi|\Psi\rangle|$. Let $M :=$ floor$(a^2 2^n)$. Then for any positive integer n and any $\delta > 0$ there exists a quantum channel that implements δ-approximately in diamond norm the map:

$$|0^n\rangle\langle 0^n| \otimes |\Psi\rangle\langle\Psi| \to (p\,|M\rangle\langle M| + (1-p)\,|M-1 \textit{ mod } 2^n\rangle\langle M-1 \textit{ mod } 2^n|) \otimes |\Psi\rangle\langle\Psi| \tag{214}$$

for some probability p (i.e., there is some probability that instead of obtaining floor$(a^2 2^n)$ we obtain floor$(a^2 2^n) - 1$). This channel uses $O\left(2^n \log(\delta^{-1})\right)$ controlled applications of R_Π and $R_{|\Psi\rangle} := 2|\Psi\rangle\langle\Psi| - I$.

Proof. We use linear combinations of unitaries to make block-encodings of Π and $|\Psi\rangle\langle\Psi|$.

$$\Pi = \frac{I + R_\Pi}{2}, \qquad |\Psi\rangle\langle\Psi| = \frac{I + R_{|\Psi\rangle}}{2} \tag{215}$$

Then, we multiply these projectors together to make a block-encoding of A:

$$A := |\Psi\rangle\langle\Psi| \cdot \Pi \cdot |\Psi\rangle\langle\Psi| = a^2 |\Psi\rangle\langle\Psi| \tag{216}$$

Now we simply invoke the algorithm described in Corollary 16 for the case when no rounding promise is present, and apply it to the input state $|\Psi\rangle$. Since $|\Psi\rangle$ is an eigenstate of A, we are guaranteed to measure either floor$(a^2 2^{n-1})$ or floor$(a^2 2^{n-1}) - 1 \bmod 2^n$. □

Observe that the random error only occurs for particular values of a^2, which is where the rounding promise is violated. We can ignore the rounding promise precisely because the input state $|\Psi\rangle$ is an eigenstate of the Hamiltonian: an incorrect estimate does not damage the input state.

6 Acknowledgments

The author thanks Scott Aaronson, Andrew Tan, Pawel Wocjan, William Zeng, Yosi Atia, András Gilyén, Philip Jensen, Lasse Bjørn Kristensen and Alán Aspuru-Guzik for helpful conversations. This work was supported by Scott Aaronson's Vannevar Bush Faculty Fellowship from the US Department of Defense.

References

[WT21] Pawel Wocjan, Kristan Temme **Szegedy Walk Unitaries for Quantum Maps** arXiv:2107.07365 (2021)

[MRTC21] John M. Martyn, Zane M. Rossi, Andrew K. Tan, Isaac L. Chuang **A Grand Unification of Quantum Algorithms** arXiv:2105.02859 (2021)

[LT21] Lin Lin, Yu Tong **Heisenberg-limited ground state energy estimation for early fault-tolerant quantum computers** arXiv:2102.11340 (2021)

[Cam20] Earl T. Campbell **Early fault-tolerant simulations of the Hubbard model** arXiv:2012.09238 (2020)

[SHC20] Yuan Su, Hsin-Yuan Huang, Earl T. Campbell **Nearly tight Trotterization of interacting electrons** arXiv:2012.09194 Quantum 5, 495 (2020)

[ESP20] Alexander Engel, Graeme Smith, Scott E. Parker **A framework for applying quantum computation to nonlinear dynamical systems** arXiv:2012.06681 Physics of Plasmas 28, 062305 (2020)

[An&20] Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, Jiasu Wang **Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance** arXiv:2012.06283 Quantum 5, 481 (2020)

[Chuang20] Isaac Chuang. **Grand unification of quantum algorithms.** Seminar presentation at IQC Waterloo. (2020)

[Wright&20] Lewis Wright, Fergus Barratt, James Dborin, George H. Booth, Andrew G. Green **Automatic Post-selection by Ancillae Thermalisation** arXiv:2010.04173 Phys. Rev. Research 3, 033151 (2020)

[AHNTW20] Srinivasan Arunachalam, Vojtech Havlicek, Giacomo Nannicini, Kristan Temme, Pawel Wocjan **Simpler (classical) and faster (quantum) algorithms for Gibbs partition functions** arXiv:2009.11270 (2020)

[GST20] András Gilyén, Zhao Song, Ewin Tang **An improved quantum-inspired algorithm for linear regression** arXiv:2009.07268 (2020)

[JKKA20] Phillip W. K. Jensen, Lasse Bjørn Kristensen, Jakob S. Kottmann, Alán Aspuru-Guzik **Quantum Computation of Eigenvalues within Target Intervals** Quantum Science and Technology 6, 015004 arXiv:2005.13434 (2020)

[Rall20] Patrick Rall **Quantum Algorithms for Estimating Physical Quantities using Block-Encodings** Phys. Rev. A 102, 022408 arXiv:2004.06832 (2020)

[Rogg20] Alessandro Roggero **Spectral density estimation with the Gaussian Integral Transform** Phys. Rev. A 102, 022409 arXiv:2004.04889 (2020)

[Chao&20] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, Mario Szegedy **Finding Angles for Quantum Signal Processing with Machine Precision** arXiv:2003.02831 (2020)

[LT21] Lin Lin, Yu Tong **Near-optimal ground state preparation** arXiv:2002.12508 Quantum 4, 372 (2020)

[Childs&20] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, Shuchen Zhu **A Theory of Trotter Error** Phys. Rev. X 11, 011020 arXiv:1912.08854 (2019)

[GGZW19] Dmitry Grinko, Julien Gacon, Christa Zoufal, Stefan Woerner **Iterative Quantum Amplitude Estimation** npj Quantum Inf 7, 52 arXiv:1912.05559 (2019)

[Lemi&19] Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore, Matthias Troyer **Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm** Quantum 4, 287 arXiv:1910.01659 (2019)

[AR19] Scott Aaronson, Patrick Rall. **Quantum Approximate Counting,Simplified** Symposium on Simplicity in Algorithms. 2020, 24-32 arXiv:1908.10846(2019)

[HW19] Aram W. Harrow, Annie Y. Wei. **Adaptive Quantum Simulated Annealing for Bayesian Inference and Estimating Partition Functions** Proc. of SODA 2020 arXiv:1907.09965 (2019)

[KLLP18] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash **q-means: A quantum algorithm for unsupervised machine learning** arXiv:1812.03584 NIPS 32 (2018)

[HM18] Yassine Hamoudi, Frédéric Magniez. **Quantum Chebyshev's Inequality and Applications** ICALP, LIPIcs Vol 132, pages 69:1-99:16 arXiv:1807.06456 (2018)

[Haah18] Jeongwan Haah **Product Decomposition of Periodic Functions in Quantum Signal Processing** Quantum 3, 190. arXiv:1806.10236 (2018)

[GSLW18] András Gilyén, Yuan Su, Guang Hao Low, Nathan Wiebe **Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics** arXiv:1806.01838 Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019) Pages 193–204 (2018)

[Poulin&17] David Poulin, Alexei Kitaev, Damian S. Steiger, Matthew B. Hastings, Matthias Troyer **Quantum Algorithm for Spectral Measurement with Lower Gate Count** arXiv:1711.11025 Phys. Rev. Lett. 121, 010501 (2017)

[LC17] Guang Hao Low, Isaac L. Chuang **Hamiltonian Simulation by Uniform Spectral Amplification** arXiv:1707.05391 (2017)

[KP17] Iordanis Kerenidis, Anupam Prakash **Quantum gradient descent for linear systems and least squares** arXiv:1704.04992 Phys. Rev. A 101, 022316 (2017)

[AA16] Yosi Atia, Dorit Aharonov **Fast-forwarding of Hamiltonians and exponentially precise measurements** Nature Communications volume 8, 1572 arXiv:1610.09619 (2016)

[LC1610] Guang Hao Low, Isaac L. Chuang **Hamiltonian Simulation by Qubitization** Quantum 3, 163 arXiv:1610.06546 (2016)

[LC1606] Guang Hao Low, Isaac L. Chuang **Optimal Hamiltonian Simulation by Quantum Signal Processing** Phys. Rev. Lett. 118, 010501 arXiv:1606.02685 (2016)

[KP16] Iordanis Kerenidis, Anupam Prakash **Quantum Recommendation Systems** arXiv:1603.08675 ITCS 2017, p. 49:1–49:21 (2016)

[CKS15] Andrew M. Childs, Robin Kothari, Rolando D. Somma **Quantum algorithm for systems of linear equations with exponentially improved dependence on precision** SIAM Journal on Computing 46, 1920-1950 arXiv:1511.02306 (2015)

[Mon15] Ashley Montanaro **Quantum speedup of Monte Carlo methods** Proc. Roy. Soc. Ser. A, vol. 471 no. 2181, 20150301 arXiv:1504.06987 (2015)

[KLY15] Shelby Kimmel, Guang Hao Low, Theodore J. Yoder **Robust Calibration of a Universal Single-Qubit Gate-Set via Robust Phase Estimation** Phys. Rev. A 92, 062315 arXiv:1502.02677 (2015)

[BCK15] Dominic W. Berry, Andrew M. Childs, Robin Kothari **Hamiltonian simulation with nearly optimal dependence on all parameters** arXiv:1501.01715 Proc. FOCS, pp. 792-809 (2015)

[TaShma13] Amnon Ta-Shma **Inverting well conditioned matrices in quantum logspace** STOC '13, Pages 881–890 (2013)

[Beals&12] Robert Beals, Stephen Brierley, Oliver Gray, Aram Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, Mark Stather **Efficient Distributed Quantum Computing** Proc. R. Soc. A 2013 469, 20120686 arXiv:1207.2307 (2012)

[ORR11] Maris Ozols, Martin Roetteler, Jérémie Roland **Quantum Rejection Sampling** arXiv:1103.2774 IRCS'12 pages 290-308 (2011)

[YA11] Man-Hong Yung, Alán Aspuru-Guzik **A Quantum-Quantum Metropolis Algorithm** arXiv:1011.1468 PNAS 109, 754-759 (2011)

[Ambainis10] Andris Ambainis **Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations** arXiv:1010.4458 STACS'12, 636-647 (2010)

[Temme&09] K. Temme, T.J. Osborne, K.G. Vollbrecht, D. Poulin, F. Verstraete **Quantum Metropolis Sampling** arXiv:0911.3635 Nature volume 471, pages 87–90 (2009)

[Diak09] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco Servedio, Emanuele Viola **Bounded Independence Fools Halfspaces** arXiv:0902.3757 FOCS '09, Pages 171–180 (2009)

[HHL08] Aram W. Harrow, Avinatan Hassidim, Seth Lloyd **Quantum algorithm for solving linear systems of equations** Phys. Rev. Lett. 103, 150502 arXiv:0811.3171 (2008)

[Higgins07] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, G. J. Pryde **Entanglement-free Heisenberg-limited phase estimation** Nature.450:393-396 arXiv:0709.2996 (2007)

[MW05] Chris Marriott, John Watrous **Quantum Arthur-Merlin Games** CC, 14(2): 122 - 152 arXiv:cs/0506068 (2005)

[Sze04] Mario Szegedy **Quantum speed-up of Markov chain based algorithms** FOCS '04, Pages 32-41 (2004)

[Klauck02] Hartmut Klauck **Quantum Time-Space Tradeoffs for Sorting** STOC 03, Pages 69–76 arXiv:quant-ph/0211174 (2002)

[HNS01] Peter Hoyer, Jan Neerbek, Yaoyun Shi **Quantum complexities of ordered searching, sorting, and element distinctness** 28th ICALP, LNCS 2076, pp. 346-357 arXiv:quant-ph/0102078 (2001)

[NC00] Isaac Chuang and Michael Nielsen **Quantum Computation and Quantum Information** Cambridge University Press. ISBN-13: 978-1107002173 (2000)

[BHMT00] Gilles Brassard, Peter Hoyer, Michele Mosca, Alain Tapp **Quantum Amplitude Amplification and Estimation** Quantum Computation and Quantum Information, 305:53-74 arXiv:quant-ph/0005055 (2000)

[AKN98] Dorit Aharonov, Alexei Kitaev, Noam Nisan **Quantum Circuits with Mixed States** STOC '97, pages 20-30 arXiv:quant-ph/9806029 (1998)

[NW98] Ashwin Nayak, Felix Wu **The quantum query complexity of approximating the median and related statistics** arXiv:quant-ph/9804066 STOC '99 pp 384-393 (1998)

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, Umesh Vazirani **Strengths and Weaknesses of Quantum Computing** arXiv:quant-ph/9701001 SIAM Journal on Computing 26(5):1510-1523 (1997)

[Kit95] A. Yu. Kitaev **Quantum measurements and the Abelian Stabilizer Problem** arXiv:quant-ph/9511026 (1995)

[Shor95] Peter W. Shor **Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer** SIAM J.Sci.Statist.Comput. 26, 1484 arXiv:quant-ph/9508027 (1995)

[Rivlin69] Theodore J. Rivlin **An Introduction to the Approximation of Functions** Dover Publications, Inc. New York. ISBN-13:978-0486640693 (1969)