

Adaptive quantum phase estimation can be better than non-adaptive

Noah Linden*

Ronald de Wolf†

November 10, 2025

Abstract

Quantum phase estimation is one of the most important tools in quantum algorithms. It can be made non-adaptive (meaning all applications of the unitary U_ϕ happen simultaneously) without using more applications of U_ϕ , albeit at the expense of using many more qubits. It is also known that there is no advantage for adaptive algorithms in the case where the phase that needs to be estimated is arbitrary or is uniformly random. Here we give examples of a special case of phase estimation, with a promise on the values that the unknown phase can take, where adaptive methods *are* provably better than non-adaptive methods by a factor of nearly 2 in the number of uses of U_ϕ . We also prove some upper bounds on the maximum advantage that adaptive algorithms for phase estimation can achieve over non-adaptive ones.

1 Introduction

1.1 Phase estimation

In phase estimation we have the ability to apply a unitary U_ϕ (possibly controlled by another qubit) and are given a copy of an eigenstate $|\Psi\rangle$ of U_ϕ with eigenvalue $e^{2\pi i\phi}$. The task is to estimate the phase $\phi \in [0, 1)$. Kitaev [Kit95] showed how to estimate ϕ with additive error δ (modulo 1) using roughly $1/\delta$ applications of U_ϕ . The standard circuit for this is as follows. We use a set of $k = \lceil \log(1/\delta) \rceil$ auxiliary qubits starting in state $|0\rangle$ which will end up containing a k -bit estimate of ϕ . These k qubits are first put into a uniform superposition over all $j = 0, \dots, 2^k - 1$, which controls how many times U_ϕ is applied to $|\Psi\rangle$. Because $|j\rangle \otimes U_\phi^j |\Psi\rangle = e^{2\pi i j \phi} |j\rangle \otimes |\Psi\rangle$, we obtain the following k -qubit state (tensored with our one copy of $|\Psi\rangle$)

$$\frac{1}{\sqrt{2^k}} \sum_{j=0}^{2^k-1} e^{2\pi i j \phi} |j\rangle. \quad (1)$$

By writing the binary expansion of j as a sum of powers of 2, this can be nicely factored as

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^{k-1}\phi}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^{k-2}\phi}|1\rangle) \otimes \dots \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \phi}|1\rangle). \quad (2)$$

*School of Mathematics, University of Bristol. Partially supported by the UK Engineering and Physical Sciences Research Council through grant EP/T001062/1. n.linden@bristol.ac.uk

†QuSoft, CWI and University of Amsterdam, the Netherlands. Partially supported by the Dutch Research Council (NWO) through Gravitation-grant Quantum Software Consortium, 024.003.037. rdewolf@cwi.nl

One can think of this as applying the phase ϕ 2^{k-1} times on the first qubit, 2^{k-2} times on the second qubit, etc. An inverse quantum Fourier transform (QFT, denoted Q below) will then convert this state into a state whose amplitudes are concentrated around the best k -bit approximations of ϕ (mod 1). In fact, if ϕ can be written exactly with k bits of precision, then the circuit produces ϕ with probability 1, because the state of Eq. (2) is then $Q|2^k\phi\rangle$. The circuit is depicted below.

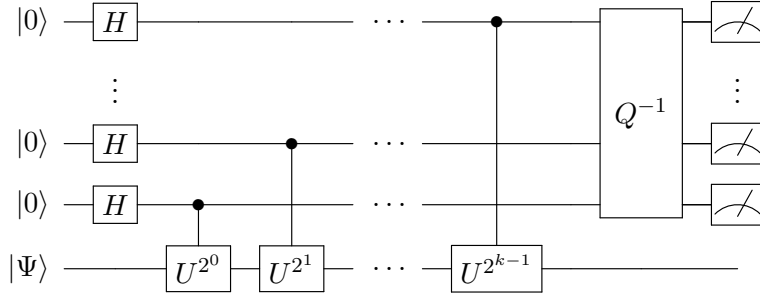


Figure 1: The standard circuit for phase estimation

Many quantum algorithms can be cast as an application of phase estimation [CEMM98], including integer factoring [Sho97, Kit95] and amplitude estimation [BHMT02] (used for approximate counting). Phase estimation is a crucial subroutine in quantum walks (used there to reflect about the uniform superposition over all edges of a graph [MNRS11]), in the negative-weights adversary bound in query complexity (used there to turn a dual-SDP solution into a quantum algorithm [Rei09, LMR⁺11]) and in many applications in chemistry and physics (used there for instance to measure the energy of a state).

A fundamental question for quantum computing is to what extent quantum algorithms can be *parallelized*. Parallelization reduces the time qubits need to stay coherent, at the expense of having to do many operations simultaneously, and typically using more auxiliary qubits.¹ We call a phase estimation algorithm *non-adaptive* if all its applications of U_ϕ are applied in parallel, meaning at the same time step. This corresponds to full parallelization, at least of the applications of U_ϕ . In contrast, in an *adaptive* algorithm earlier applications of U_ϕ can affect the state on which later U_ϕ 's are applied. For example, the standard circuit for phase estimation (Figure 1) is adaptive. The goal of this paper is to describe cases where adaptive phase estimation algorithms can be provably more efficient than non-adaptive ones.²

We note that our notion of an adaptive algorithm allows for measurements which are performed in the middle of the computation, between applications of U_ϕ , and subsequent actions of the algorithm are dependent on the classical outcomes of those measurements. However, if we only care about the number of applications of U_ϕ , then by the so-called “principle of deferred measurement” (see e.g. [NC00, Section 4.4]) we can move all measurements towards the end, making everything coherent without increasing the number of uses of U_ϕ . This works by replacing a measurement of a qubit by a CNOT gate that “copies” the qubit to an auxiliary qubit that is initially $|0\rangle$ and that is not touched further in the computation. Later parts of the computation are then conditioned

¹Thus in practical applications, low-depth algorithms can have reduced impact of noise because there is less time for the qubits to decohere.

²Our use of the term “non-adaptive” follows computer-science terminology, and should not be confused with other uses in quantum sensing [DRC17, Section VIII.A.2] and quantum information [DFLS16].

on the unmeasured qubit rather than on a classical binary measurement outcome. It is not hard to show that the final output distribution, after tracing out the auxiliary qubits, will be exactly the same as for the original algorithm that used intermediate measurements. The classical control that might select the next operation to do based on intermediate measurement outcomes can be simulated with a lot of controlled operations. This can significantly increase the total number of gates in the circuit, but not the number of applications of (controlled) U_ϕ .

1.2 A basic problem in sensing

The following is a basic problem in *quantum sensing*: estimate the parameter $\phi \in [0, 1)$ given the ability to apply a one-qubit unitary

$$U_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i \phi} \end{pmatrix}, \quad (3)$$

This is of course just the special case of phase estimation where the eigenstate $|\Psi\rangle$ is $|1\rangle$, and $|0\rangle$ is an eigenstate with eigenvalue 1 (equivalently, eigenphase 0). Note that if we can implement such a U_ϕ then we can also implement controlled- U_ϕ ourselves (by swapping the target with the eigenvalue-1 state $|0\rangle$ if the control bit is 0), so we no longer need to *assume* the ability to apply controlled version of U_ϕ . One might also hope that knowing the eigenvectors could make phase estimation cheaper, but that is not the case: the standard circuit (with $|\Psi\rangle = |1\rangle$) is still optimal for obtaining a good k -bit approximator of ϕ with high success probability, both in the worst-case and in the average-case where ϕ is uniformly random.³

In this simple case, phase estimation can be made non-adaptive without using more applications of U_ϕ than the standard circuit. The idea is to produce each one of the k qubits of the product state of Eq. (2) using its own “cat state”. This is similar to a well known approach in quantum sensing, see for instance this comprehensive review [DRC17, Sec. VIII.A & IX.C]. For each $p = 2^{k-1}, 2^{k-2}, \dots, 2, 1$ we create the p -qubit state

$$\frac{1}{\sqrt{2}}(|0^p\rangle + |1^p\rangle)$$

starting from $|0^p\rangle$ using a Hadamard and $p - 1$ CNOTs. We can then apply $U_\phi^{\otimes p}$ to induce the phase $e^{2\pi i p \phi}$ on $|1^p\rangle$, and invert the CNOTs to end with the qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i p \phi}|1\rangle)$. Doing this in parallel for each of the k qubits in the state (2), we prepare that state in a way where

³This optimality for the average case of uniformly random $\phi \in [0, 1)$ was proved by van Dam et al. [DDMM07]. It is not difficult to see that the same (average-case) proof will work for ϕ being a uniformly random integer multiple of 2^{-k} . In fact, the worst case for the latter problem can be reduced to the average case by changing the U_ϕ by adding a uniformly random offset in $\{0, 2^{-k}, 2 \cdot 2^{-k}, 3 \cdot 2^{-k}, \dots, 1 - 2^{-k}\}$ to ϕ before the algorithm (and subtracting it from the obtained estimate at the end), as is done in [LW22]. This shows that the average-case phase estimation problem is no easier than the worst-case for this set of phases. [DDMM07] considered a cost function $C(\phi, \tilde{\phi})$ between the actual unknown phase ϕ and the algorithm’s estimator $\tilde{\phi}$, under the assumption that $C(\cdot, \cdot)$ only depends on the difference $\phi - \tilde{\phi}$. They proved their result by showing how every adaptive algorithm can be massaged into a non-adaptive one with the same number of applications of U_ϕ and the same expected cost function (i.e., the expectation $\mathbb{E}[C(\phi, \tilde{\phi})]$ does not go up, where the expectation is taken over a prior distribution over ϕ and over the algorithm’s randomness in producing an estimator $\tilde{\phi}$ on input ϕ). In our paper the cost measure is the “01-loss” $C(\phi, \tilde{\phi}) = 1 - \delta_{\phi, \tilde{\phi}}$, which satisfies their assumption. However, their result does not contradict our factor-2 improvement because their proof crucially only preserves the expectation of $C(\phi, \tilde{\phi})$ under a *uniform* prior distribution over all $\phi \in [0, 1)$, while our phases ϕ come from a very specific phase-set F .

the applications of U_ϕ are all in parallel, all at the same timestep, thus “flattening” the circuit without using more applications of U_ϕ than the standard adaptive circuit.⁴ The price we pay for this parallelization is the large number $O(2^k)$ of qubits needed to prepare the cat states. This is an extreme example of a width-vs-depth tradeoff, where we greatly reduce the depth of the circuit at the expense of greatly increasing the number of qubits. Note that the circuits for preparing the cat states and for the final inverse QFT, are not constant-depth, but can be implemented in logarithmic depth; for cat states this can be done by a binary tree of CNOT gates, and for the QFT this was done by Cleve and Watrous [CW00].

1.3 Our result: adaptive phase-estimation algorithms can be twice as efficient

In many applications of phase estimation and sensing, the phase ϕ that we want to estimate is not completely arbitrary, but comes from some limited set F of possible phases. For example, it could be that the possible phases correspond to certain specific energies, like the specific spectral lines of some molecule. We want to find examples in this setting where adaptive phase estimation is provably more efficient than non-adaptive phase estimation, meaning parallelization comes at a cost in the number of applications of U_ϕ (in contrast to promiseless phase estimation, where the standard circuit can be parallelized without using more applications of U_ϕ , as we argued above).

We start in Section 2 with a simple example with 4 possible values for ϕ : $0, 2^{-k}, 1/2, 1/2 + 2^{-k+1}$. Note that the first two phases can easily be distinguished from the other two phases by looking at $U_\phi|+\rangle$, which is equal or close to $|+\rangle$ for the first two phases, and equal or close to $|-\rangle$ for the other two phases. An adaptive algorithm can make this distinction first, with error probability $O(2^{-k})$. If it finds itself in the first case, then it uses 2^{k-1} applications of U_ϕ to prepare the qubit $U_\phi^{2^{k-1}}|+\rangle$ to resolve the difference between phases 0 and 2^{-k} . In the second case it uses 2^{k-2} applications of U_ϕ to prepare the qubit $U_\phi^{2^{k-2}}|+\rangle$ to resolve the difference between $1/2$ and $1/2 + 2^{-k+1}$. So in the worst case this uses $\frac{1}{2}2^k + 1$ applications of U_ϕ to learn which of the 4 possibilities ϕ is.

This upper bound of $\frac{1}{2}2^k + 1$ applications of U_ϕ is optimal (up to lower-order terms) for distinguishing the 4 phases in F , even adaptively. In fact $(\frac{1}{2} - o(1))2^k$ applications of U_ϕ are already needed just to distinguish the phases 0 and 2^{-k} with success probability $1 - o(1)$ (where $o(1)$ denotes any function that goes to 0 with k), as follows. Up to a global phase and a change of basis, our U_ϕ can be thought of as a rotation matrix over angle $\pi\phi$. By the principle of deferred measurement, we may assume w.l.o.g. that the distinguishing circuit is unitary, i.e., no intermediate measurements (this has a cost in extra auxiliary qubits and controlled operations, but not more applications of U_ϕ). If we want to distinguish the cases $\phi = 0$ from $\phi = 2^{-k}$, then the initial state needs to be rotated to two different states at angle $\pi/2 - o(1)$, depending on which of the two cases we are in. Since each application of U_ϕ can increase the angle between the states by at most $\pi \cdot 2^{-k}$ (see the appendix), we need at least $(\pi/2 - o(1))/(\pi \cdot 2^{-k}) = (\frac{1}{2} - o(1))2^k$ applications of U_ϕ .⁵

⁴Something similar would work to “flatten” arbitrary phase estimation, with arbitrary multi-qubit U_ϕ and unknown $|\Psi\rangle$, but it would require us to prepare ket-states of the form $\frac{1}{\sqrt{2}}(|\Psi_1\rangle^{\otimes p} + |\Psi_2\rangle^{\otimes p})$ where $|\Psi_1\rangle$ and $|\Psi_2\rangle$ are eigenstates of U_ϕ whose relative phase is ϕ . This seems much more demanding than a one-qubit unitary U_ϕ as in Eq. (3).

⁵We focus on error probability $1 - o(1)$ since this allows us to get precise constants (up to additive $o(1)$) in the number of uses of U_ϕ . If, in contrast, we had wanted only constant error probability, say 0.01, then $N = 0.47 \cdot 2^k$ applications of U_ϕ would already suffice to distinguish phases 0 and 2^{-k} , by preparing the qubit $U_\phi^N|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \chi|1\rangle)$, where $\chi = 1$ if $\phi = 0$ and $\chi = e^{2\pi i \cdot 0.47} \approx -0.982 + i0.187$ if $\phi = 2^{-k}$. Doing a Hadamard gate and measuring in the computational basis gives outcome 0 with certainty if $\phi = 0$, and gives outcome 1 with probability $\approx |(1.982 - i0.187)/2|^2 \approx 0.991$ if $\phi = 2^{-k}$.

Hence our algorithm is essentially optimal for this 4-phase task. This algorithm is adaptive: it first (cheaply) distinguishes the first pair of phases from the second pair, and then based on that outcome determines the next phase estimation it will perform.

How well can a *non-adaptive* algorithm do for distinguishing these 4 phases? Initially one might expect that such an algorithm has to separately do the work for recovering the k th and the $(k-1)$ th bits of ϕ , since it cannot adaptively find out first which of these two is needed, and the qubit $U_\phi^{2^{k-1}}|+\rangle$ that is used to learn the k th bit of ϕ gives no information about the $(k-1)$ th bit of ϕ . The number of applications of U_ϕ would then be $2^{k-1} + 2^{k-2} = \frac{3}{4}2^k$. However, we will show here that in fact $(\frac{2}{3} \pm o(1))2^k$ applications of U_ϕ are necessary and sufficient for non-adaptive phase estimation with error probability $1 - o(1)$. This is better than the naive non-adaptive algorithm, but still a factor $4/3$ worse than the number of applications of U_ϕ that suffice for an adaptive algorithm.

Our upper bound proof for the number of uses of U_ϕ in the non-adaptive case finds explicit amplitudes for the initial state on which the U_ϕ 's are applied in parallel. Our lower bound proof massages the existence of a non-adaptive algorithm with N applications of U_ϕ into a linear constraint problem. Using Farkas's lemma we can then show the *infeasibility* of that problem: obtaining a lower bound on the number of applications of U_ϕ by exhibiting a feasible solution for the *dual* problem. This approach via duality to rule out a suitable initial state is (to the best of our knowledge) a novel proof method to show the non-existence of a good quantum algorithm.

In Section 3 we extend the 4-phase example to larger sets of phases. In particular, by choosing a set F_ℓ that consists of 2^ℓ pairs of phases, where the two elements of each pair are separated by a different multiple of 2^{-k} (e.g., $2^{-k}, 2 \cdot 2^{-k}, 3 \cdot 2^{-k}, \dots$), we construct an instance of phase estimation where roughly $\frac{1}{2}2^k$ applications of U_ϕ still suffice for an adaptive algorithm, while for non-adaptive algorithms, $\frac{2^\ell}{2^\ell+1}2^k$ applications of U_ϕ are necessary and sufficient. For large constant ℓ , this gives an advantage of nearly a factor of 2 for adaptive algorithms over non-adaptive ones.

As we discuss in Section 4, these separations of a factor of 2 are essentially best-possible for the type of sets F that we consider here.

1.4 Other examples of adaptive advantage

We give in this paper a small adaptive-vs-non-adaptive advantage for phase estimation. This advantage is one of many examples of adaptive advantage in quantum computing, and more broadly in other branches of computing as well. It is not surprising that for some computational problems parallelization is not possible without significant overhead, leaving us with an adaptive advantage, though this is not always easy to prove. As a famous example, the Euclidean algorithm is an efficient adaptive algorithm for computing the greatest common divisor of two integers in polynomial time, and it is a longstanding conjecture that this algorithm cannot be parallelized to logarithmic depth.

Other settings in quantum computing allow for much larger (exponential) adaptive advantage, for example quantum communication complexity, based on versions of “pointer jumping” functions. There the input is a tree of instances of some smaller communication complexity problem, each internal problem instance's solution points to one of its children. These pointers chart a length- k path from the root of the tree to a particular leaf, which then contains the solution to the overall problem. For problem-trees of depth k , k -round interactive communication protocols can just adaptively follow the length- k path to the correct leaf; on the other hand, one can show that one-way (or even $(k-1)$ -round) quantum communication protocols need exponentially more com-

munication [KNTZ07]. One can similarly (and much more easily) construct exponential adaptive separations in adaptive vs. non-adaptive quantum *query* complexity.

In adaptive vs. non-adaptive *channel discrimination*, [HHLW10] gave a pair of entanglement-breaking channels that can be perfectly discriminated adaptively by two channel evaluations, but for which every non-adaptive strategy that evaluates the channel a finite number of times has some positive error probability. More recently, [SHW22] showed that for discriminating two classical-quantum channels, adaptive and non-adaptive strategies lead to the same error exponents in various asymptotic settings, while for discriminating fully quantum channels adaptivity can lead to a better exponent. In the non-asymptotic regime with asymptotically vanishing type I error probability, [BDSW24] showed that adaptive channel discrimination strategies can be made non-adaptive at the expense of a small loss in the error exponent and a small increase in the number of uses of the channel.

2 First example: a set F with 4 possible phases

2.1 The problem and an upper bound of $\frac{1}{2}2^k$ for adaptive algorithms

In this section we make our first example precise. We consider the problem of phase estimation with a single-qubit unitary U_ϕ as in Eq. (3), with the phase ϕ coming from the set

$$F = \{0, 2^{-k}, 1/2, 1/2 + 2^{-k+1}\}. \quad (4)$$

As we already argued in the introduction, an *adaptive* algorithm can identify ϕ with error probability $O(2^{-k})$ using $\frac{1}{2}2^k + 1$ applications of U_ϕ .

2.2 An upper bound of $\frac{2}{3}2^k$ for non-adaptive algorithms

Because the 4 phases in the set F are either close to or equal to 0, or close to or equal to $1/2$, we can learn the most significant bit of ϕ with only one application of U_ϕ , by just preparing $U_\phi|+\rangle$ and measuring in the $|\pm\rangle$ basis. If $\phi \in \{0, 2^{-k}\}$ then we will almost certainly (with error probability $\exp(-k)$) get outcome $|+\rangle$, and if $\phi \in \{1/2, 1/2 + 2^{-k+1}\}$ then we will almost certainly get outcome $|-\rangle$. We can also non-adaptively learn the k th and $(k-1)$ th bits of ϕ with success probability 1, by preparing $U_\phi^{\frac{1}{2}2^k}|+\rangle$ and $U_\phi^{\frac{1}{4}2^k}|+\rangle$ and doing a joint measurement on those 2 qubits. This non-adaptive algorithm uses $\frac{1}{2}2^k + \frac{1}{4}2^k + 1 = \frac{3}{4}2^k + 1$ applications of U_ϕ .

However we will now describe a more efficient non-adaptive algorithm which uses roughly $\frac{2}{3}2^k$ applications of U_ϕ , and which identifies $\phi \in F$ with error probability $\exp(-k)$. An arbitrary non-adaptive algorithm with N applications of U_ϕ would prepare some initial state

$$|\psi\rangle = \sum_{x \in \{0,1\}^N} \alpha_x |x\rangle |\eta_x\rangle \quad (5)$$

of $q \geq N$ qubits, apply $U_\phi^{\otimes N} \otimes I_{2^{q-N}}$ (which adds phase $e^{2\pi i|x|\phi}$ to $|x\rangle$, where $|x|$ is the Hamming weight of x), and do a measurement with $|F|$ outcomes corresponding to the possible phases $\phi \in F$.

We will now massage such an arbitrary non-adaptive protocol into a form that is a bit easier to analyse. For $j = 0, \dots, N$, define the orthonormal states $|\chi_j\rangle = \frac{1}{\beta_j} \sum_{x \in \{0,1\}^N: |x|=j} \alpha_x |x\rangle |\eta_x\rangle$,

where $\beta_j = \sqrt{\sum_{x \in \{0,1\}^N: |x|=j} |\alpha_x|^2}$ is a normalizing factor. Then we can write the state before the measurement as

$$\sum_{j=0}^N \beta_j e^{2\pi i j \phi} |\chi_j\rangle \quad (6)$$

By doing a unitary that is independent of ϕ , we may assume the state before the measurement is actually the $(N+1)$ -dimensional state

$$|\psi_\phi\rangle = \sum_{j=0}^N \beta_j e^{2\pi i j \phi} |j\rangle. \quad (7)$$

A non-adaptive algorithm that uses U_ϕ a total of N times, is thus fully described by the unit vector of amplitudes $(\beta_0, \dots, \beta_N)$.

We will now show how to choose the β_j 's to obtain a non-adaptive algorithm for learning phases $\phi \in F$ with roughly $\frac{2}{3}2^k$ applications of U_ϕ . Our non-adaptive algorithm will have two registers: one containing one qubit, and one containing an $(N+1)$ -dimensional state ($\lceil \log N \rceil$ qubits). The total number of applications of U_ϕ will be $N+1$: we act with U_ϕ on the first qubit and with the map $|j\rangle \mapsto e^{2\pi i j \phi} |j\rangle$ on the second register (which corresponds to the action of $U_\phi^{\otimes N} \otimes I$ on the original initial state of Eq. (5)) to produce the final two-register state $|\psi_\phi\rangle$. We want the four states $|\psi_0\rangle, |\psi_{2^{-k}}\rangle, |\psi_{1/2}\rangle, |\psi_{1/2+2^{-k+1}}\rangle$ to be almost orthogonal, so we can distinguish them with high success probability using some specific measurement. Specifically, we choose N to be a multiple of 4 closest to $\frac{2}{3}2^k$, thus $N = \frac{2}{3}2^k + \Delta$, where $|\Delta| < 2$, and we choose the initial state $|\psi\rangle$ to be

$$|\psi\rangle = |+\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|N/2\rangle + \frac{1}{\sqrt{3}}|N\rangle \right). \quad (8)$$

Because we chose N a multiple of 4, the most-significant bit of ϕ (i.e., whether ϕ is close to 0 or to $1/2$) is irrelevant if we apply the phase 0 or $N/2$ or N times.

Thus

$$\begin{aligned} |\psi_0\rangle &= |+\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|N/2\rangle + \frac{1}{\sqrt{3}}|N\rangle \right) \\ |\psi_{2^{-k}}\rangle &= |+\rangle' \left(\frac{1}{\sqrt{3}}|0\rangle + \eta \frac{1}{\sqrt{3}}|N/2\rangle + \eta^2 \frac{1}{\sqrt{3}}|N\rangle \right) \\ |\psi_{1/2}\rangle &= |-\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|N/2\rangle + \frac{1}{\sqrt{3}}|N\rangle \right) \\ |\psi_{1/2+2^{-k+1}}\rangle &= |-\rangle' \left(\frac{1}{\sqrt{3}}|0\rangle + \eta^2 \frac{1}{\sqrt{3}}|N/2\rangle + \eta^4 \frac{1}{\sqrt{3}}|N\rangle \right), \end{aligned} \quad (9)$$

where

$$\eta = \exp 2\pi i \frac{N}{2^{k+1}}. \quad (10)$$

The states $|+\rangle' = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i/2^k}|1\rangle)$ and $|-\rangle' = \frac{1}{\sqrt{2}}(|0\rangle - e^{2\pi i/2^{k-1}}|1\rangle)$ are exponentially (in k) close to $|+\rangle$ and $|-\rangle$ respectively. Since we chose $N = \frac{2}{3}2^k + \Delta$ with $|\Delta| < 2$, we see that η is exponentially close to a third root of unity:

$$\eta = \exp \left(2\pi i \left(\frac{1}{3} + \frac{\Delta}{2^{k+1}} \right) \right). \quad (11)$$

Thus, defining ω as the third root of unity $\omega = \exp \frac{2}{3}\pi i$, the states $|\psi_0\rangle, |\psi_{2^{-k}}\rangle, |\psi_{1/2}\rangle, |\psi_{1/2+2^{-k+1}}\rangle$ are exponentially close to

$$\begin{aligned} &|+\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|N/2\rangle + \frac{1}{\sqrt{3}}|N\rangle \right) \\ &|+\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \omega \frac{1}{\sqrt{3}}|N/2\rangle + \omega^2 \frac{1}{\sqrt{3}}|N\rangle \right) \\ &|-\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|N/2\rangle + \frac{1}{\sqrt{3}}|N\rangle \right) \\ &|-\rangle \left(\frac{1}{\sqrt{3}}|0\rangle + \omega^2 \frac{1}{\sqrt{3}}|N/2\rangle + \omega \frac{1}{\sqrt{3}}|N\rangle \right), \end{aligned} \quad (12)$$

respectively. We can now measure the first qubit in the Hadamard basis and the second register with a projective measurement that distinguishes the three pairwise-orthogonal states of the second register. If we had one of the 4 states of Eq. (12), then this measurement would give the correct outcome with probability 1. Since the state we are actually measuring is exponentially (in k) close to one of these 4 states, we will get the correct outcome except with exponentially small probability.

2.3 A lower bound of $\frac{2}{3}2^k$ for non-adaptive algorithms

We note that the adaptive algorithm in Section 1.3 achieves error probability that goes to 0 with k . For fair comparison, we require that the non-adaptive algorithm also has error probability $\kappa = o(1)$ (meaning κ goes to 0 with k). Here we will prove that every non-adaptive algorithm needs at least $(\frac{2}{3} - o(1))2^k$ applications of U_ϕ to achieve $\kappa = o(1)$. Thus our algorithm in the previous section is essentially best possible. This proves a constant-factor advantage of adaptive over non-adaptive phase-estimation algorithms: at most $\frac{1}{2}2^k + 1$ vs. more than $(\frac{2}{3} - o(1))2^k$ applications of U_ϕ .

Recall from the previous section that a non-adaptive algorithm with N applications of U_ϕ , is fully described by the $(N+1)$ -dimensional final states

$$|\psi_\phi\rangle = \sum_{j=0}^N \beta_j e^{2\pi i j \phi} |j\rangle. \quad (13)$$

A non-adaptive algorithm with (worst-case) error probability κ will (among other things) have to distinguish phase $\phi = 2^{-k}$ from phase $\phi' = 0$, and also distinguish phase $\phi = 1/2 + 2^{-k+1}$ from $\phi' = 1/2$. This implies that the two complex inner products

$$\langle \psi_0 | \psi_{2^{-k}} \rangle = \sum_{j=0}^N \beta_j^2 e^{2\pi i j / 2^k} \quad \text{and} \quad \langle \psi_{1/2} | \psi_{1/2+2^{-k+1}} \rangle = \sum_{j=0}^N \beta_j^2 e^{4\pi i j / 2^k} \quad (14)$$

both have small magnitude: at most $O(\sqrt{\kappa})$, since the probability to “confuse” two states corresponds to their squared inner product. Translating to 4 real equations, there must exist small real

numbers $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ (possibly negative) of magnitude $O(\sqrt{\kappa})$ such that

$$\sum_{j=0}^N \cos(2\pi j/2^k) \beta_j^2 = \varepsilon_1 \quad (15)$$

$$\sum_{j=0}^N \sin(2\pi j/2^k) \beta_j^2 = \varepsilon_2 \quad (16)$$

$$\sum_{j=0}^N \cos(4\pi j/2^k) \beta_j^2 = \varepsilon_3 \quad (17)$$

$$\sum_{j=0}^N \sin(4\pi j/2^k) \beta_j^2 = \varepsilon_4. \quad (18)$$

Let A be the $5 \times (N+1)$ real matrix whose first row is all-1s, and whose last 4 rows correspond to the sines and cosines in the above 4 equations, respectively. Let E be the column vector $(1, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)^T$. If there is a non-adaptive algorithm, then there will exist a vector $b \in \mathbb{R}^{N+1}$ (corresponding to the β_j^2 , which sum to 1) satisfying

$$Ab = E \text{ and } b \geq 0. \quad (19)$$

Farkas's lemma [Far02] (which may be found in any text on linear programming or convex geometry) says that the system (19) is *unsatisfiable* iff the following system over variable-vector y is *satisfiable*

$$y^T A \geq 0 \text{ and } y^T E < 0. \quad (20)$$

Geometrically, this lemma says that either the vector E is in the positive cone spanned by the columns of A (with b being the vector of positive coefficients), or there is a hyperplane that separates E from that positive cone (in the sense that the hyperplane's normal vector y has positive inner product with each of the columns of A but negative inner product with E).

Accordingly, we can use Farkas's lemma to lower bound the cost of non-adaptive algorithms by just exhibiting 5 numbers: if we can find a $y \in \mathbb{R}^5$ satisfying the latter system, then we have proved that every non-adaptive algorithm that can identify the phases in F with error probability $o(1)$, needs to use more than N applications of U_ϕ .

Our strategy for exhibiting such a 5-dimensional vector will be as follows. Firstly, we will find a setting of the y_j 's with $y_0 = 0$ that satisfies the first inequality in (20). But this choice does not necessarily satisfy the second inequality of (20). We then choose y_0 to be a small negative value, in order to satisfy the second inequality; but we will also add small amounts to the other y_j 's (depending on y_0) to ensure that the first inequality is still satisfied; we will arrange things so that this change to the y_j 's for $j \neq 0$ is sufficiently small that both inequalities in (20) are satisfied, with the previously chosen value of y_0 .

In detail this works as follows in this case. Towards the goal of finding a suitable y -vector, define the real function

$$f(x) = y_0 + y_1 \cos(2\pi x) + y_2 \sin(2\pi x) + y_3 \cos(4\pi x) + y_4 \sin(4\pi x). \quad (21)$$

Note that $y^T A \geq 0$ is equivalent to

$$f(x) \geq 0 \text{ for all } x \in \{0, \dots, N\}/2^k. \quad (22)$$

We will now exhibit values for the y -vector that satisfy a stronger condition than above, namely

$$f(x) \geq 0 \text{ for all } x \in [0, N/2^k]. \quad (23)$$

First, consider the case where $(y_0, y_1, y_2, y_3, y_4) = (0, -1/4, \sqrt{3}/4, 1/4, \sqrt{3}/4)$. We will first analyse this choice of y (with $y_0 = 0$) and show that the induced function f satisfies the inequality (23). Then, following the strategy described in the previous paragraph but one, we make y_0 negative in order to also satisfy the constraint $y^T E < 0$, while adding a small function to f (slightly changing the other y_j 's) in order to preserve the validity of (23) for an interval of x whose right boundary is only slightly below $N/2^k$.

Using trigonometric identities one may write the particular instantiation of the function f as follows (where the subscript in f_1 below indicates $\ell = 1$, which will be generalized to larger ℓ in the next section)

$$f_1(x) = -\frac{1}{4} \cos(2\pi x) + \frac{\sqrt{3}}{4} \sin(2\pi x) + \frac{1}{4} \cos(4\pi x) + \frac{\sqrt{3}}{4} \sin(4\pi x) = \sin(3\pi x) \sin(\pi x + \frac{2\pi}{3}). \quad (24)$$

Thus we may see from the right-hand side that $f_1(x)$ has zeros at 0, 1/3, and 2/3, and $f_1(x)$ is positive in between (since its two factors have the same sign between the zeros), satisfying Eq. (23) for $N = \frac{2}{3}2^k$. This is illustrated in Figure 2.

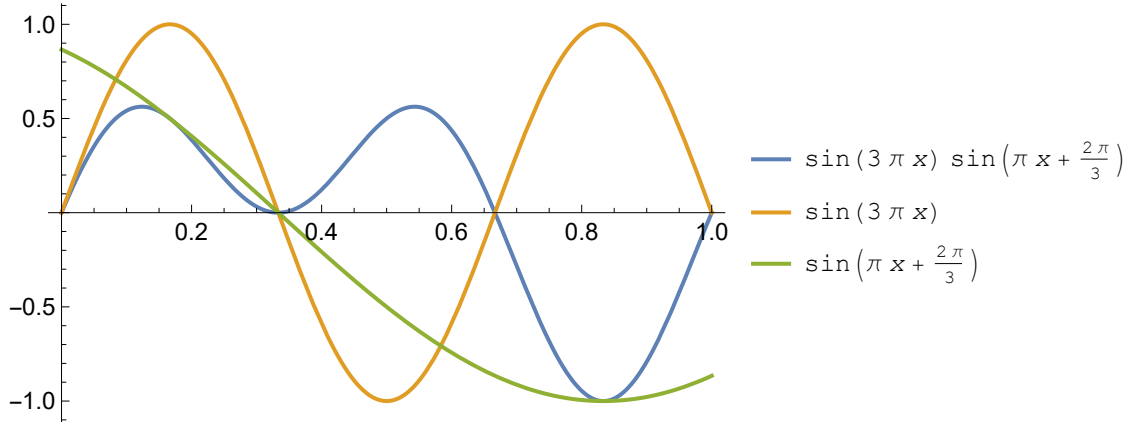


Figure 2: The function $f_1(x)$ (in blue) and the two functions that make its product form

This $y \in \mathbb{R}^5$ does not necessarily satisfy the constraint $y^T E < 0$ of Eq. (20) because $y_0 = 0$. So we now choose y_0 to be sufficiently negative, setting it to $y_0 = -2(|\varepsilon_1| + |\varepsilon_2| + |\varepsilon_3| + |\varepsilon_4|) - c(k)$, where $c(k) = o(1)$ is a strictly positive function. This implies

$$y^T E = y_0 + \sum_{j=1}^4 y_j \varepsilon_j \leq -(|\varepsilon_1| + |\varepsilon_2| + |\varepsilon_3| + |\varepsilon_4|) - c(k) < 0, \quad (25)$$

where we used that our chosen y_1, y_2, y_3, y_4 all have magnitude bounded below 1.

However the resulting function $f_1(x) + y_0$ has value $y_0 < 0$ at the zeros of f_1 , so the condition of Eq. (23) doesn't hold anymore in an interval of width $o(1)$ around the zeros of f_1 . But we can

modify the function by one final step to make it positive again at these points. We now add a small amount of the function $\sin(2\pi x + \pi/6)$. Specifically we form the function

$$\tilde{f}_1(x) = f_1(x) + y_0 + 4|y_0|\sin(2\pi x + \pi/6). \quad (26)$$

The key point is that $\sin(2\pi x + \pi/6)$ equals $1/2$ at $x = 0$ and $x = 1/3$, and is at least $1/4$ in constant-sized intervals around $x = 0$ and $x = 1/3$. Hence adding $4|y_0|\sin(2\pi x + \pi/6)$ to $f_1(x) + y_0$ lifts the function above zero, compensating for the fact that y_0 is negative, and ensuring that the modified function $\tilde{f}_1$ is positive everywhere between $x = 0$ and some point close to $2/3$. The function $\sin(2\pi x + \pi/6)$ is also expressible as a linear combination of $\cos(2\pi x)$ and $\sin(2\pi x)$, so we change y_1 and y_2 by an amount with magnitude $O(|y_0|) = o(1)$ to obtain $\tilde{f}_1$. The inequality Eq. (25) remains valid because our modified y_1 and y_2 still have magnitude ≤ 1 , so now we have satisfied both constraints of Eq. (20).

This modification of f_1 to $\tilde{f}_1$ will reduce the value of x near $x = 2/3$ where the new function is zero (because $\sin(2\pi x + \pi/6)$ is negative around $x = 2/3$), but only by an amount that can be made as small as we like, because $y_0 = o(1)$. This proves that the linear system Eq. (20) is satisfiable for $N = \gamma \cdot 2^k$, where γ can be made as close as we like to $2/3$; and therefore Eq. (19) is unsatisfiable. Hence we have proved a lower bound of $(\frac{2}{3} - o(1))2^k$ applications of U_ϕ that holds for every non-adaptive algorithm that can distinguish the 4 phases in the set F with error probability $o(1)$.

3 A factor-2 advantage from larger phase-sets

3.1 The general problem and an upper bound of $\frac{1}{2}2^k$ for adaptive algorithms

Let us now consider a larger set of possible phases. The previous set F had 2 pairs of phases, one pair close to 0 at distance 2^{-k} , and one pair close to $1/2$ at distance $2 \cdot 2^{-k}$. We will now consider a set

$$F_\ell \text{ with } 2^\ell \text{ pairs of phases, one pair } r2^{-\ell}, r2^{-\ell} + (r+1)2^{-k}, \text{ for each } r = 0, 1, \dots, 2^\ell - 1. \quad (27)$$

For example, the previous 4-phase set is F_1 , and F_2 is the 8-phase set

$$\{0, 2^{-k}, 1/4, 1/4 + 2 \cdot 2^{-k}, 1/2, 1/2 + 3 \cdot 2^{-k}, 3/4, 3/4 + 4 \cdot 2^{-k}\} \quad (28)$$

Each phase $\phi \in F$ is a k -bit number whose leading (i.e., most-significant) ℓ bits determine to which of the 2^ℓ pairs the phase belongs; the last (i.e., least-significant) ℓ bits correspond to $(r+1)2^{-k}$; and in between those two ℓ -bit strings is a sequence of $k - 2\ell$ 0s. Assuming ℓ is constant and $\ell \ll k$, there still is an adaptive algorithm for phase estimation with this phase set that uses $\frac{1}{2}2^k + O(2^\ell) \approx \frac{1}{2}2^k$ applications of U_ϕ , as follows. First estimate the first ℓ bits of ϕ cheaply, using $O(2^\ell)$ applications of U_ϕ . This has error probability $O(2^{-k})$ thanks to the fact that there is a long sequence of at least $k - 2\ell = k - O(1)$ 0s in ϕ following its first ℓ bits. Learning the ℓ most significant-bits narrows down the possible phases to one pair $r2^{-\ell}, r2^{-\ell} + (r+1)2^{-k}$. Second, use at most $\frac{1}{2}2^k$ applications of U_ϕ to learn (with success probability 1) the least-significant bit of ϕ . That bit is 1 for the phase $r2^{-\ell} + (r+1)2^{-k}$, and 0 for the phase $r2^{-\ell}$, so learning that one bit suffices to distinguish the two phases in the remaining pair.

3.2 An upper bound of $\frac{2^\ell}{2^\ell+1}2^k$ for non-adaptive algorithms

We claim that there is a non-adaptive algorithm with essentially $\frac{2^\ell}{2^\ell+1}2^k$ applications of U_ϕ to distinguish these phases. This is essentially optimal for $\ell = 1$ as we showed above (via Farkas's lemma), and for $\ell \geq 2$ as we will show below in Section 3.3.

First, using $2^\ell - 1$ applications of U_ϕ we can learn the ℓ most-significant bits of ϕ by standard phase estimation “flattened out” on many qubits to make it non-adaptive (as explained in Section 1.2). Once we know which multiple $r2^{-\ell}$ of $2^{-\ell}$ our unknown phase ϕ is close (or equal) to, it remains to distinguish the two phases $r2^{-\ell}$ and $r2^{-\ell} + (r+1)2^{-k}$, i.e., to resolve a phase difference $(r+1)2^{-k}$. The crucial thing that distinguishes non-adaptive from adaptive algorithms here, is that a non-adaptive algorithm cannot use a state that depends on r : it has to use the same β -vector of amplitudes for all r .

Let N be $\frac{2^\ell}{2^\ell+1}2^k$, rounded to the nearest multiple of 2^{2^ℓ} . The rounding induces a shift $\Delta = N - \frac{2^\ell}{2^\ell+1}2^k$ of magnitude $|\Delta| < 2^{2^\ell-1}$ compared to the intended number $\frac{2^\ell}{2^\ell+1}2^k$; it ensures that $N/2^\ell$ is an integer multiple of 2^ℓ . Now consider the following state, which is supported on integer multiples of $N/2^\ell$:

$$\frac{1}{\sqrt{2^\ell+1}} \sum_{m=0}^{2^\ell} \left| \frac{mN}{2^\ell} \right\rangle. \quad (29)$$

The largest integer in this superposition is indeed N , which is $\frac{2^\ell}{2^\ell+1}2^k + \Delta$, and which corresponds to the number of applications of U_ϕ of the non-adaptive algorithm. Note that the state of Eq. (8) is the $\ell = 1$ case, where N is $\frac{2}{3}2^k$ rounded to the nearest multiple of 4.

If we apply $|j\rangle \mapsto e^{2\pi i j \phi} |j\rangle$, then we obtain the state

$$\frac{1}{\sqrt{2^\ell+1}} \sum_{m=0}^{2^\ell} e^{2\pi i m N \phi / 2^\ell} \left| \frac{mN}{2^\ell} \right\rangle, \quad (30)$$

Note that because $N/2^\ell$ is an integer multiple of 2^ℓ , the phases in this superposition do not depend on the first ℓ bits of ϕ . To show that this non-adaptive algorithm works, it now suffices to show near-orthogonality for the states induced by phases 0 and $(r+1)2^{-k}$, for each $r = 0, \dots, 2^\ell - 1$, because then there exists a measurement that distinguishes all $|F|$ states with exponentially small error probability (using the first part of the state to learn the first ℓ bits of ϕ and the state of (30) to resolve the difference between the pair of phases induced by the first ℓ bits). The inner product between the two states for phases $(r+1)2^{-k}$ and 0 is

$$\frac{1}{2^\ell+1} \sum_{m=0}^{2^\ell} e^{2\pi i m N (r+1) 2^{-k} / 2^\ell} = \frac{1}{2^\ell+1} \sum_{m=0}^{2^\ell} e^{2\pi i m ((r+1)/(2^\ell+1) + \Delta((r+1)2^{-k})/2^\ell)}, \quad (31)$$

where the equality used that $N = \frac{2^\ell}{2^\ell+1}2^k + \Delta$. We have $m\Delta((r+1)2^{-k})/2^\ell \leq 2^{3\ell-1-k}$ (because $m \leq 2^\ell, \Delta \leq 2^{2^\ell-1}, r \leq 2^\ell - 1$), which is an exponentially small (in k) change to the phase $e^{2\pi i m (r+1)/(2^\ell+1)}$ because we think of ℓ as a constant and k growing. Without that small change in the phase, the geometric sum $\sum_{m=0}^{2^\ell} e^{2\pi i m (r+1)/(2^\ell+1)} = 0$. Hence the inner product of (31) is exponentially small (in k).

3.3 A lower bound of $\frac{2^\ell}{2^\ell+1}2^k$ for non-adaptive algorithms

We now extend the argument for the lower bound of Section 2.3 to $\ell \geq 2$. We again give a lower bound via Farkas's lemma, by providing a function that is positive for $x \in [0, \frac{2^\ell}{2^\ell+1}]$ and that is a sum of sines and cosines of the form

$$\sum_{j=1}^{2^\ell} \left(y_{2j-1} \cos(2\pi jx) + y_{2j} \sin(2\pi jx) \right), \quad (32)$$

where the coefficients $y_1, \dots, y_{2^\ell+1}$ are real.

Here is the particular function we will consider

$$f_\ell(x) = \sin \left(\pi \left(x - \frac{1}{2^\ell+1} \right) \right) \sin \left(\pi (2^\ell+1)x \right) \prod_{m=2}^{2^{\ell-1}} \left(\cos(2\pi \frac{m}{2^\ell+1}) - \cos(2\pi x) \right). \quad (33)$$

We will show that this function is positive for $x \in [0, \frac{2^\ell}{2^\ell+1}]$, and, in the region $x \in [0, 1]$ only has zeros at integer multiples of $1/(2^\ell+1)$. To understand why this function satisfies our desiderata, it will be instructive to first consider a particular case, for example $\ell = 3$:

$$f_3(x) = \sin \left(\pi \left(x - \frac{1}{9} \right) \right) \sin(9\pi x) \prod_{m=2}^4 \left(\cos(2\pi \frac{m}{9}) - \cos(2\pi x) \right). \quad (34)$$

The function $f_3(x)$ together with the five component functions in the product form are illustrated in Figures 3 and 4, respectively.

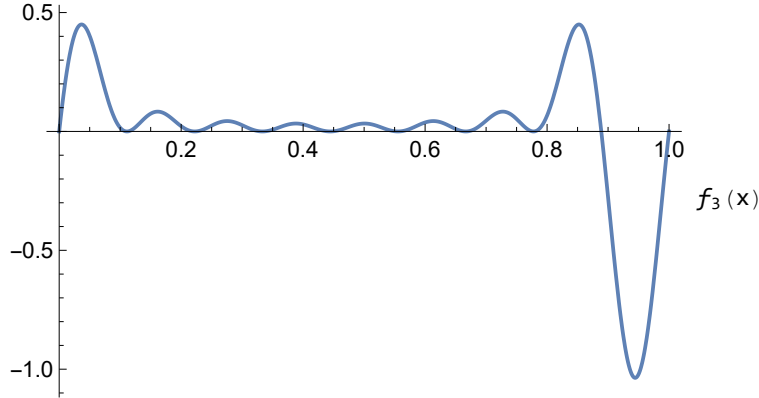


Figure 3: $f_3(x)$ as defined in (34)

We are only interested in the region $x \in [0, 1]$, so in what follows, any statements only concern values x in this region. In order to understand why $f_3(x)$ is positive on $[0, 8/9]$, it is helpful to consider the signs of the component functions in the 9 regions $[1, 1/9], [1/9, 2/9], \dots, [8/9, 1]$.

Firstly, we note that $\sin(9\pi x)$ alternates in sign in each successive region of width $1/9$: its sign pattern is $(+, -, +, -, +, -, +, -, +)$.

Then consider the function

$$g_m(x) = \cos(2\pi \frac{m}{9}) - \cos(2\pi x) \quad (35)$$

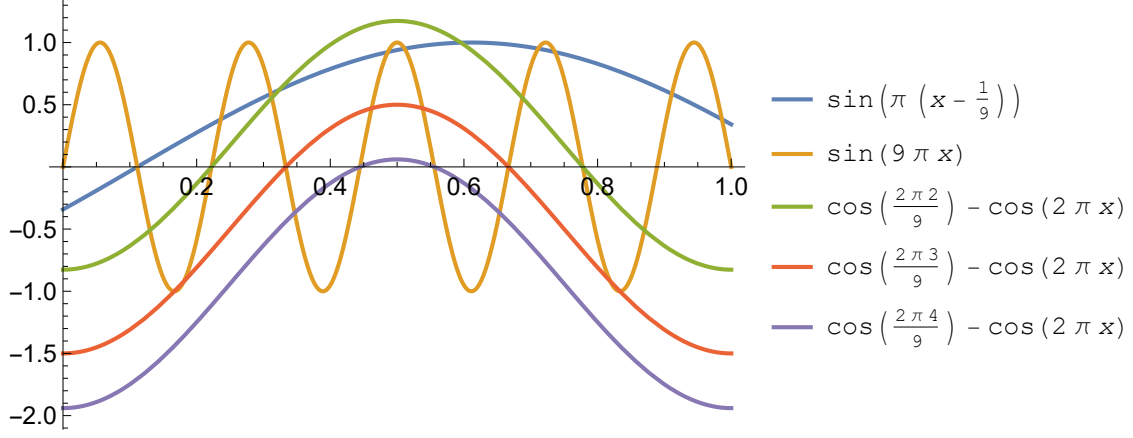


Figure 4: The five functions in the product form for $f_3(x)$

for $m = 2, 3, 4$. In the region $x \in [0, 1]$, $g_m(x)$ only has zeros at $x = m/9$ and $x = 1 - m/9$, and it is positive for $x \in [m/9, 1 - m/9]$ and negative otherwise. Thus, if we consider each successive region of length $1/9$, the functions g_2, g_3, g_4 have sign patterns, $(-, -, +, +, +, +, +, -)$, $(-, -, -, +, +, +, -, -)$, $(-, -, -, -, +, -, -, -)$, respectively.

Thus the product

$$\prod_{m=2}^4 \left(\cos(2\pi \frac{m}{9}) - \cos(2\pi x) \right) \quad (36)$$

has sign pattern $(-, -, +, -, +, -, +, -)$, and therefore the product

$$\sin(9\pi x) \prod_{m=2}^4 \left(\cos(2\pi \frac{m}{9}) - \cos(2\pi x) \right) \quad (37)$$

has sign pattern $(-, +, +, +, +, +, +, -)$. Lastly, $\sin(\pi(x - \frac{1}{9}))$ is negative for $x \in [0, 1/9]$ and positive otherwise, so its sign pattern is $(-, +, +, +, +, +, +, +)$. Accordingly, for each of the first 8 regions, multiplying out the signs of the five component functions gives $+$, so $f_3(x)$ is positive in $x \in [0, 8/9]$, as required.

Furthermore (for example by writing each cosine and sine using $\cos a = (e^{ia} + e^{-ia})/2$ and $\sin a = (e^{ia} - e^{-ia})/2i$) it may be seen that $f_3(x)$ only contains the required cosines and sines as in Eq. (32).

Just as in the case $\ell = 1$ in Section 2.3, we need to modify f_3 to make sure both parts of the requirements of Farkas's lemma (Eq. (20)) are satisfied. In particular, the constraint $y^T E < 0$ is not guaranteed to hold for the choice of y_j 's (incl. $y_0 = 0$) that define the current f_3 . We again make two modifications. Firstly we add a small negative y_0 ; namely let $Y_{\max}$ be the maximum value of the modulus of the y_j , for $j \neq 0$.⁶ Then we take

$$y_0 = -2Y_{\max} \sum_{j=1}^{2^{\ell}+1} |\epsilon_j|. \quad (38)$$

⁶We note that a simple upper bound for f_3 is 2^3 . This upper bounds (using triangle inequality) the y_j since they are the Fourier coefficients of f_3 when decomposing f_3 as a linear combination of sines and cosines; hence in fact $Y_{\max} \leq 2^4$, but all that is needed for our purposes is that it is upper bounded by a constant as $k \rightarrow \infty$.

Secondly, the addition of this negative y_0 means the function $f_3 + y_0$ will be negative around the zeros of f_3 , so we need to add a function to $f_3 + y_0$ to make sure we get a function that is positive in the entire required region. Fortunately, there is a simple choice at hand: using similar arguments to the ones we have just made for f_3 , the function f_2 is strictly positive at the places where f_3 is zero in the region $x \in [0, 8/9]$. (The zeros of f_2 are at integer multiples of $1/5$, so they do not coincide with the zeros of f_3 ; and f_2 is positive for $x \in [0, 4/5]$.) Thus we can add a small multiple of f_2 — shifted slightly to the left (by $1/90$, which is half the minimal distance between the zeros of f_3 and the zeros of f_2) — to compensate for the addition of the small negative y_0 . This shifted function f_2 will be strictly positive at the zeros of f_3 . Let $z_{\min}$ denote the minimum value of the shifted f_2 at these zeros. Because f_2 has bounded derivative, it will then have value at least $z_{\min}/2$ in constant-width intervals around the zeros of f_3 .

Thus our modified function

$$\tilde{f}_3(x) = f_3(x) + y_0 + \frac{2|y_0|}{z_{\min}} f_2(x + 1/90) \quad (39)$$

will be positive from $x = 0$ almost up to $x = 8/9$, and contain only the allowed sines and cosines. This ensures that the first inequality in Farkas's lemma ($y^T A \geq 0$) holds for our new coefficients y_j . The coefficients y_j for $\tilde{f}_3$ are slightly different from those for f_3 ; but since $|y_0| = o(1)$ (i.e., y_0 goes to 0 as $k \rightarrow \infty$), and $z_{\min}$ and all the y_j for f_2 are independent of k , the second inequality in Farkas's lemma ($y^T E < 0$) will still hold. This was the reason for having the factor 2 in Eq. (38).

As in the case of $\ell = 1$ in Section 2.3, this addition of y_0 and f_2 will slightly reduce the region of x for which the function is positive, but only by an amount that we can make as small as we like by starting with a sufficiently small error probability κ , since the changes to the y_j are $O(\sqrt{\kappa})$ and $\kappa = o(1)$.

The general argument about the positivity of $f_\ell(x)$ for ℓ other than 3, proceeds in the same way. As before we are only interested in the region $x \in [0, 1]$. The factors after the product symbol in Eq. (33) are positive in the region $x \in [m/(2^\ell + 1), 1 - m/(2^\ell + 1)]$ and negative otherwise. So we consider regions of x of width $1/(2^\ell + 1)$. The product of these functions has the same pattern of signs as $\sin(\pi(2^\ell + 1)x)$, apart from in the first region $x \in [0, 1/(2^\ell + 1)]$, but the remaining factor in Eq. (33), namely $\sin\left(\pi\left(x - \frac{1}{2^\ell + 1}\right)\right)$, is negative in $x \in [0, 1/(2^\ell + 1)]$, and positive otherwise, so that $f_\ell(x)$ is indeed positive for all $x \in [0, 2^\ell/(2^\ell + 1)]$. And $f_\ell(x)$ only contains the required cosines and sines as in Eq. (32). Finally, as in the discussion for $\ell = 3$, we can arrange both requirements of Farkas's lemma to be satisfied by adding a small negative y_0 and by adding a small amount of a shifted version of $f_{\ell-1}$ to f_ℓ , noting that $f_{\ell-1}$ and f_ℓ have no common zeros in $[0, 1]$ ⁷, and the region in which $f_{\ell-1}$ is positive extends beyond $x \in [0, (2^\ell - 1)/(2^\ell + 1)]$.⁸

4 Are bigger separations possible?

Our examples naturally lead to the question whether there could be bigger separations between the number of applications of U_ϕ in adaptive and non-adaptive phase estimation, significantly bigger than our factor of 2.

⁷Such a common zero in $[0, 1)$ would have to be an integer multiple of $1/(2^\ell + 1)$ and of $1/(2^{\ell-1} + 1)$, but that doesn't exist because the numbers $2^\ell + 1$ and $2^{\ell-1} + 1$ are coprime (because if $p > 1$ is a factor of $2^{\ell-1} + 1$, then $(2^\ell + 1)/p = 2(2^{\ell-1} + 1)/p - 1/p$ is an integer minus $1/p$, which is not an integer).

⁸The size of y_0 and the “small amount” of $f_{\ell-1}$ could grow very rapidly with ℓ but this is not an issue since we think of ℓ as a constant and $k \rightarrow \infty$.

If all phases in F can be written as k -bit numbers, and the smallest difference $\delta = \min\{|\phi - \phi'| \bmod 1 \mid \text{distinct } \phi, \phi' \in F\}$ between any two phases in F is $\delta = 2^{-k}$ (as it is in the specific sets F_ℓ that we use for our separations), then we can show there is no separation larger than factor-2, as follows. On the one hand, the parallelized version of standard k -bit phase estimation non-adaptively recovers the phase ϕ *with success probability 1* using $2^k - 1 < 1/\delta$ applications of U_ϕ . But on the other hand, even just to resolve the difference between the minimizing phases ϕ and ϕ' , an adaptive phase estimation algorithm needs at least $(1/2 - o(1))/\delta$ applications of U_ϕ if we want to have error probability $o(1)$, by the lower bound argument of Section 1.3. So for large constant ℓ our separations based on the sets F_ℓ approach the maximal separation (i.e., factor-2) that is possible for sets F where k bits of precision are necessary and sufficient.

What about phase sets F where $\delta \approx 2^{-k}$ but some of the phases $\phi \in F$ need more than k bits of precision? The lower bound of $(1/2 - o(1))/\delta \approx \frac{1}{2}2^k$ applications of U_ϕ for adaptive algorithms that have error probability $o(1)$, remains valid. Non-adaptively, we can still do parallelized phase estimation with k bits of precision. This will use $2^k - 1$ applications of U_ϕ , and will have some *constant* error probability but not necessarily error probability $o(1)$. Hence we cannot conclude now that factor-2 separation is best possible when comparing adaptive and non-adaptive algorithms that have vanishing error probability. In general one can show that the standard phase estimation algorithm with t bits of precision has additive error $\leq 2^{-k}$ in its estimate, except with error probability $O(2^{k-t})$, see for instance [NC00, Eq. (5.35)]. We can ensure that the non-adaptive phase estimation has error probability $o(1)$ by using $t = k + b(k)$ bits of precision, for any function $b(k)$ that goes to infinity, even extremely slowly.⁹ However, the number of applications of U_ϕ of this non-adaptive phase estimation now would be $2^{k+b(k)} - 1$, and the ratio with the adaptive lower bound of roughly $\frac{1}{2}2^k$ would go to infinity with k , albeit arbitrarily slowly. It is tempting to conjecture that there exists some universal constant C (possibly even $C = 2$) such that the adaptive advantage is never more than factor- C for any phase-set F . However, we were not able to prove this and leave it as a question for future work.

5 Conclusion and discussion

We studied phase estimation on a single-qubit phase-gate U_ϕ (which is a basic scenario in quantum sensing) in the setting where we have a promise on the values that the unknown phase ϕ can take. This twist on the phase-estimation problem is possibly of some practical interest in situations where we have some prior information about the value of ϕ . We provided examples where adaptive algorithms had an advantage of nearly a factor of 2 over non-adaptive algorithms in terms of the number of applications of U_ϕ needed—in contrast to the promiseless scenario, where there is no such advantage. However we also proved that such an advantage cannot be very large: at most a factor of 2 when k bits of precision are necessary and sufficient for the promised phases, and at most an arbitrarily-slowly growing function in general (it would be very interesting to close that gap). We introduced some new techniques here that may find use elsewhere, for instance the use of Farkas’s lemma to lower bound the required number of non-adaptive uses of U_ϕ : disproving the existence of a suitable initial amplitude-vector by exhibiting a solution to a dual problem.

As mentioned in the introduction, one of the main motivations for studying fully parallel (= non-adaptive) algorithms is that they are less sensitive to noise on the qubits due to their lower depth.

⁹For instance we could use $b(k) = \log^*(k)$, which is defined as the number of times the binary logarithm needs to be applied to k iteratively before the value drops below 1.

Hence a next step would be to study phase estimation in the presence of various kinds of noise, and to study to what extent quantum advantages (both adaptive and non-adaptive) disappear. There is already a significant literature studying in what situations the desired Heisenberg scaling (where $O(1/\delta)$ applications of U_ϕ suffice to estimate $\phi \pm \delta$) gives way to so-called “classical” scaling (where $\Omega(1/\delta^2)$ applications of U_ϕ are needed), see for instance [DDKG12]. One could similarly ask to what extent adaptive-over-non-adaptive advantages survive in the presence of various kinds of noise.

Acknowledgments. Mathematica [Wol25] was used for the figures in Sections 2.3 and 3.3, and for algebraic experiments earlier in the work. We thank Ashley Montanaro for providing us with Figure 1.

References

- [BDSW24] Bjarne Bergh, Nilanjana Datta, Robert Salzmänn, and Mark M. Wilde. Parallelization of adaptive quantum channel discrimination in the non-asymptotic regime. *IEEE Transactions on Information Theory*, 70(4):2617–2636, 2024. arXiv:2206.08350.
- [BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.
- [CEMM98] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. In *Proceedings of the Royal Society of London*, volume A454, pages 339–354, 1998. quant-ph/9708016.
- [CW00] Richard Cleve and John Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proceedings of 41st IEEE FOCS*, pages 526–536, 2000. quant-ph/0006004.
- [DDKG12] Rafał Demkowicz-Dobrzański, Jan Kołodyński, and Mădălin Guță. The elusive Heisenberg limit in quantum-enhanced metrology. *Nature Communications*, 3:1063, 2012. arXiv:1201.3940.
- [DDMM07] Wim van Dam, G. Mauro D’Ariano, Chiara Macchiavello, and Michele Mosca. Optimal quantum circuits for general phase estimation. *Physical Review Letters*, 98(090501), 2007. quant-ph/0609160.
- [DFLS16] Frédéric Dupuis, Serge Fehr, Philippe Lamontagne, and Louis Salvail. Adaptive versus non-adaptive strategies in the quantum setting with applications. In *Proceedings of CRYPTO*, volume 3, pages 33–59, 2016. arXiv:1607.08168.
- [DRC17] C. L. Degen, F. Reinhard, and P. Cappellaro. Quantum sensing. *Reviews of Modern Physics*, 89:035002, 2017. arXiv:1611.02427.
- [Far02] Gyula Farkas. Theorie der Einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik*, 124:1–27, 1902.

- [HHLW10] Aram W. Harrow, Avinatan Hassidim, Debbie W. Leung, and John Watrous. Adaptive versus nonadaptive strategies for quantum channel discrimination. *Physical Review A*, 81(3):032339, 2010. arXiv:0909.0256.
- [Kit95] Alexey Yu. Kitaev. Quantum measurements and the Abelian stabilizer problem. quant-ph/9511026, 12 Nov 1995.
- [KNTZ07] Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in quantum communication. *IEEE Transactions on Information Theory*, 53(6):1970–1982, 2007. Earlier version in STOC’01. quant-ph/0603135.
- [LMR⁺11] Troy Lee, Rajat Mittal, Ben Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of 52nd IEEE FOCS*, pages 344–353, 2011. arXiv:1011.3020.
- [LW22] Noah Linden and Ronald de Wolf. Average-case verification of the quantum Fourier transform enables worst-case phase estimation. *Quantum*, 6(872), 2022. arXiv:2109.10215.
- [MNRS11] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. Earlier version in STOC’07. quant-ph/0608026.
- [NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [Rei09] Ben Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proceedings of 50th IEEE FOCS*, pages 544–551, 2009.
- [Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS’94. quant-ph/9508027.
- [SHW22] Farzin Salek, Masahito Hayashi, and Andreas Winter. Usefulness of adaptive strategies in asymptotic quantum channel discrimination. *Physical Review A*, 105:022419, 2022. arXiv:2011.06569.
- [Wol25] Wolfram Research. Mathematica, Version 14, 2025. Champaign, IL.

A Maximum angle that controlled- U_ϕ can induce

Given two quantum states $|a\rangle$ and $|b\rangle$ we define the angle $\theta \in [0, \pi/2]$ between them by

$$|\langle a|b\rangle| = \cos \theta. \quad (40)$$

Let $|\Psi\rangle$ be a normalised state of $2 + q$ qubits. We can write

$$|\Psi\rangle = |00\rangle|\psi_{00}\rangle + |01\rangle|\psi_{01}\rangle + |10\rangle|\psi_{10}\rangle + |11\rangle|\psi_{11}\rangle. \quad (41)$$

So

$$\begin{aligned} C(U_\phi) \otimes I|\Psi\rangle &= |00\rangle|\psi_{00}\rangle + |01\rangle|\psi_{01}\rangle + |10\rangle|\psi_{10}\rangle + e^{2\pi i\phi}|11\rangle|\psi_{11}\rangle \\ &:= |\eta\rangle + e^{2\pi i\phi}|11\rangle|\psi_{11}\rangle. \end{aligned} \quad (42)$$

We want the maximum angle that the controlled U_ϕ can induce, which is equivalent to minimizing the inner product between $|\Psi\rangle$ and $C(U_\phi) \otimes I|\Psi\rangle$

$$\begin{aligned} \min_{|\Psi\rangle} |\langle\Psi|C(U_\phi) \otimes I|\Psi\rangle|^2 &= \min_{|\Psi\rangle} \left| \langle\eta|\eta\rangle + e^{2\pi i\phi}\langle\psi_{11}|\psi_{11}\rangle \right|^2 \\ &= \min_{z \in [0,1]} \left| 1 - z + ze^{2\pi i\phi} \right|^2, \end{aligned} \quad (43)$$

where we have defined $\langle\psi_{11}|\psi_{11}\rangle = z$. Thus

$$\begin{aligned} \min_{|\Psi\rangle} |\langle\Psi|C(U_\phi) \otimes I|\Psi\rangle|^2 &= \min_{z \in [0,1]} 1 + 2z(1-z)(\cos(2\pi\phi) - 1) \\ &= 1 + \frac{1}{2}(\cos(2\pi\phi) - 1) = \cos^2(\pi\phi). \end{aligned} \quad (44)$$

Hence the maximum angle induced is no more than $\pi\phi$.

B Triangle inequality for angles between states

Consider three states $|a\rangle, |b\rangle, |c\rangle$, with inner products

$$\langle a|b\rangle = e^{i\theta_1} \cos \alpha, \quad \langle b|c\rangle = e^{i\theta_2} \cos \beta, \quad \langle c|a\rangle = e^{i\theta_3} \cos \gamma. \quad (45)$$

We choose $0 \leq \alpha, \beta, \gamma \leq \pi/2$. We will show that $\gamma \leq \alpha + \beta$.

We note that we could choose phases for $|a\rangle, |b\rangle, |c\rangle$ to make the phases of two of the inner products zero, but not all three.¹⁰

The Gram matrix¹¹ $G_{jk} = \langle\psi_j|\psi_k\rangle$ of any set of states $\{|\psi_j\rangle\}$ is positive semi-definite, so in particular its determinant is ≥ 0 . The determinant of the Gram matrix of the states $|a\rangle, |b\rangle, |c\rangle$ is

$$1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2 \cos \alpha \cos \beta \cos \gamma \cos(\theta_1 + \theta_2 + \theta_3). \quad (46)$$

Thus

$$1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2 \cos \alpha \cos \beta \cos \gamma \geq 0. \quad (47)$$

But

$$1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2 \cos \alpha \cos \beta \cos \gamma = 4 \sin(p) \sin(p - \alpha) \sin(p - \beta) \sin(p - \gamma), \quad (48)$$

where $p = \frac{1}{2}(\alpha + \beta + \gamma)$.

Now $\sin p \geq 0$. Without loss of generality, let us take the angles ordered as $\alpha \leq \beta \leq \gamma$. In order for the right-hand side of (48) to be positive, either (a) $p \geq \alpha$ and $p \leq \beta, \gamma$, or (b) $p \geq \gamma$ (and hence $p \geq \alpha, \beta$). In the former case, however $p \leq \beta, \gamma$ implies that $\alpha \leq 0$, contrary to our assumptions about α (other than the trivial case of $\alpha = 0$). Thus we are left with case (b), i.e. $(\alpha + \beta + \gamma)/2 \geq \gamma$, in other words $\alpha + \beta \geq \gamma$ as required.

¹⁰For example by noting that $\langle a|b\rangle\langle b|c\rangle\langle c|a\rangle = \text{Tr}(|a\rangle\langle a| \cdot |b\rangle\langle b| \cdot |c\rangle\langle c|)$ is invariant under choice of phases.

¹¹We have adapted the argument that is given for real vectors in <https://math.stackexchange.com/questions/4670434/does-angle-between-vectors-satisfy-triangle-inequality>