# An Optimal Linear-combination-of-unitaries-based Quantum Linear System Solver

SANDER GRIBLING, Université Paris Cité, Paris, France
IORDANIS KERENIDIS, Université Paris Cité and QC Ware, Paris, France
DÁNIEL SZILÁGYI, Université Paris Cité, Paris, France

Solving systems of linear equations is one of the most important primitives in many different areas, including in optimization, simulation, and machine learning. Quantum algorithms for solving linear systems have the potential to provide a quantum advantage for these problems.

In this work, we recall the Chebyshev iterative method and the corresponding optimal polynomial approximation of the inverse. We show that the Chebyshev iteration polynomial can be efficiently evaluated both using quantum singular value transformation (QSVT) as well as linear combination of unitaries (LCU). We achieve this by bounding the 1-norm of the coefficients of the polynomial expressed in the Chebyshev basis. This leads to a considerable constant-factor improvement in the runtime of quantum linear system solvers that are based on LCU or QSVT (or, conversely, a several orders of magnitude smaller error with the same runtime/circuit depth).

CCS Concepts: • **Theory of computation** → **Quantum computation theory;**

Additional Key Words and Phrases: Linear systems, quantum algorithms, complexity

**ACM Reference Format:**
Sander Gribling, Iordanis Kerenidis, and Dániel Szilágyi. 2024. An Optimal Linear-combination-of-unitaries-based Quantum Linear System Solver. *ACM Trans. Quantum Comput.* 5, 2, Article 9 (April 2024), 23 pages.
https://doi.org/10.1145/3649320

## 1 INTRODUCTION

The **quantum linear systems (QLS)** problem asks for a state that encodes the solution of a linear system $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$.[1] Solving linear systems of equations appears as a subproblem in many downstream applications in optimization and in machine learning. Some examples include least-squares regression [6], support-vector machines [17, 27], as well as differential equations [19, 31]. Thus, optimizing the resources (depth, in particular) required for solving QLS would bring us closer to running these algorithms on near-term quantum hardware.

In a seminal work, Reference [15] showed how to solve the QLS-problem using only polylog$(n)$ queries to the input. Their algorithm has a polynomial dependence on the condition number $\kappa$ of $A$ and the desired precision $\varepsilon > 0$. Subsequent work has improved the $\kappa$-dependence to near

[1]Without loss of generality, one may assume that $A$ is Hermitian.

Authors' addresses: S. Gribling, I. Kerenidis, and D. Szilágyi, Université Paris Cité, Paris, France, F-75013; e-mail: gribling@irif.fr, jkeren@irif.fr, dszilagyi@irif.fr.
Current Author's address: S. Gribling, Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands; e-mail: s.j.gribling@tilburguniversity.edu.

ACM 2643-6817/2024/04-ART9
https://doi.org/10.1145/3649320

linear [1],[2] and the error-dependence to polylog$(1/\varepsilon)$ [8]. The algorithms in References [1, 8, 15] can all be viewed as implementing a polynomial transformation of $A$ that approximates the inverse. They are based on various combinations of Hamiltonian simulation, quantum walks, linear combinations of unitaries, and most recently the **quantum singular value transformation framework (QSVT)** [13, 21].

Very recently, the QLS-problem has been studied using an adiabatic approach [2, 9, 18, 30]. Interestingly, the state-of-the-art QLS-algorithm now comes from the adiabatic framework: In Reference [9] the complexity of QLS is improved from $O(\kappa \log(\kappa/\varepsilon))$ (achieved using polynomial-based techniques) to $O(\kappa \log(1/\varepsilon))$. Their algorithm prepares a low-precision estimate of the $|A^{-1}\mathbf{b}\rangle$ using a gate-based implementation of the discrete adiabatic evolution and improves its quality using quantum eigenstate filtering.

In this work, we revisit the polynomial-based QLS solvers and show that the optimal approximation polynomial (arising from *Chebyshev iteration* [26, 32]) can be efficiently evaluated using a quantum algorithm. In recent years, two quantum-algorithmic methods have been developed for evaluating polynomials at matrices (see Section 3 for more details): the **linear combination of unitaries (LCU)** lemma [5] and the QSVT framework [13]. The latter is asymptotically optimal (see Reference [13, Theorem 73]), but requires the computation of certain angles (see Section 3.1 for details), and doing so efficiently in a numerically stable way is the subject of ongoing research [7, 10, 14]. Moreover, the QSVT framework applies to polynomials that are bounded in absolute value by 1 on the interval $[-1, 1]$. Showing such a bound for polynomials $p$ that approximate functions $f : [-1, 1] \to [-1, 1]$ is straightforward. However, when $f$ is only defined on a subset of $[-1, 1]$ it is less so. One way to show that an approximating polynomial $p$ is bounded is to express $p$ in a basis of polynomials that are bounded on $[-1, 1]$ (e.g., the monomial basis, or the Chebyshev basis) and bound the 1-norm of the coefficients. The LCU approach takes such a decomposition, unitaries to implement the basis elements, and produces $p(A)$ as a suitable linear combination of said unitaries. The LCU approach is perhaps simpler, but trades that simplicity for a logarithmic overhead in the resources (multiplicative in the depth and additive in the number of qubits) as well as a subnormalization that is directly proportional to the 1-norm of the polynomial coefficients. Therefore, the motivating question is to determine which polynomials can be efficiently evaluated using LCU (at the cost of only a log-overhead).

The *asymptotically* best polynomial-based QLS-algorithm uses a polynomial introduced by Reference [8] (abbreviated as CKS from now on), which yields a natural LCU-based algorithm. In a nutshell, the CKS polynomial is obtained by starting from the polynomial $p_t(x) := \frac{1-(1-x^2)^t}{x}$ for $t = \widetilde{O}(\kappa^2)$, expressing it in the Chebyshev basis, and truncating the sum after $\widetilde{O}(\kappa)$ terms. Since the Chebyshev coefficients can be interpreted as probabilities of a certain binomial distribution, their 1-norm is easily bounded by a constant.

Our main technical contribution is to show that the Chebyshev iteration polynomial also has a bounded Chebyshev coefficient 1-norm (see Theorem 17) and can thus be evaluated using LCU. Additionally, the same norm bound implies that the polynomial can be efficiently evaluated using QSVT, yielding an *optimal* (as opposed to *asymptotically* optimal) QLS algorithm within the framework of polynomial-based solvers.

In more detail, our approach is as follows: First recall that the Chebyshev iteration corresponds to the polynomial

$$q_t(x) := \frac{\mathcal{T}_t\left(\frac{1+1/\kappa^2-2x^2}{1-1/\kappa^2}\right)/\mathcal{T}_t\left(\frac{1+1/\kappa^2}{1-1/\kappa^2}\right)}{x},$$

[2]Here, by a near-linear runtime in terms of $\kappa$, we mean a runtime that scales as $\kappa$ polylog$(\kappa)$.

where $\mathcal{T}_t$ is the $t$th Chebyshev polynomial of the first kind. These Chebyshev polynomials are defined as $\mathcal{T}_0(x) = 1$, $\mathcal{T}_1(x) = x$, and $\mathcal{T}_{t+1}(x) = 2x\mathcal{T}_t(x) - \mathcal{T}_{t-1}(x)$ for $t \geq 1$. They have the property that $|\mathcal{T}_t(x)| \leq 1$ for all $x \in [-1, 1]$ and $t \geq 0$. One can show that the polynomial $q_t$ is an $\varepsilon$-approximation of the inverse on the domain $x \in [-1, -1/\kappa] \cup [1/\kappa, 1]$, whenever $t \geq \frac{1}{2}\kappa \log(2\kappa^2/\varepsilon)$. To bound the maximum absolute value of $q_t$ on $[-1, 1]$, we express $q_t(x)$ as $\sum_{i=0}^{t-1} c_i \mathcal{T}_{2i+1}(x)$ and bound the 1-norm of the vector $\mathbf{c}$. The vector of coefficients can be used to implement $q_t(A)/\|\mathbf{c}\|_1$ either directly via the LCU approach or via the QSVT approach.

In Appendix A, we show that this approach of bounding the 1-norm of the vector of coefficients in the Chebyshev basis more generally leads to near optimal quantum algorithms via the LCU framework for a variety of continuous functions (powers of monomials, exponentials, logarithms) and discontinuous functions (the error function and by extension the sign and rectangle functions). For these functions, the coefficient norm is only a logarithmic factor away from the maximum absolute value on the interval $[-1, 1]$, meaning that they can be approximately evaluated with LCU in addition to QSVT, with slightly deeper circuits (multiplicative logarithmic overhead) and slightly more qubits (additive logarithmic overhead).

The state-of-the-art quantum linear systems solvers have a complexity that grows linearly in the condition number $\kappa$. In the small-$\kappa$ regime ($\kappa = O(n)$), it has long been known that $\Omega(\kappa)$ queries to the entries of the matrix are also needed for general linear systems [15], and recently this bound has (surprisingly) been extended to the case of positive definite systems [24]. For larger $\kappa$ less is known. For example, we do not know if quantum algorithms can improve classical algorithms if $\kappa$ is large (i.e., can we beat matrix multiplication time?). We do not even have a linear lower bound: Are $\Omega(n^2)$ queries needed when $\kappa = \Omega(n^2)$? In Reference [11] this question was answered positively when one wants to obtain a *classical description* of $A^{-1}\mathbf{b}$, and here we present a simplified proof of this result.

*Organization.* In Section 2, we recall the different approximation polynomials used for approximately solving linear systems. In Section 3, we recall the LCU and QSVT algorithms for evaluating matrix polynomials on a quantum computer. The main technical result of the article is contained in Section 4, where we show that the Chebyshev iteration polynomial can be efficiently evaluated using LCU and QSVT. We present some numerical evidence for the improved efficiency in Section 5. Finally, in Section 6, we give an overview of known lower bounds on the complexity of quantum linear system solvers both in the small $\kappa$ regime and in the large $\kappa$ regime.

## 2 PRELIMINARIES

We first provide some general references. Our approach is based on polynomial approximations to the inverse function. For an overview of fast classical algorithms using such an approach, see, for example, References [29, 33], which both discuss iterative methods to solve linear systems from an optimization perspective. For more information about Chebyshev polynomials, see, for example, Reference [28].

### 2.1 Polynomials and Approximations

*Problem definition.* We consider linear systems that are defined by a Hermitian $n$-by-$n$ matrix $A \in \mathbb{C}^{n\times n}$ and a unit vector $\mathbf{b} \in \mathbb{C}^n$. We use $\kappa$ to denote a guaranteed upper bound on the condition number of $A$, that is, we assume that all non-zero eigenvalues of $A$ lie in the set $D_\kappa := [-1, -1/\kappa] \cup [1/\kappa, 1]$. Our goal is to *approximately* solve the linear system

$$A\mathbf{x} = \mathbf{b}.$$

One can consider different notions of approximate solutions. Two natural ones are the following:

(1) return $\tilde{\mathbf{x}}$ such that $\|\tilde{\mathbf{x}} - A^{-1}\mathbf{b}\| \leq \varepsilon$.
(2) return $\tilde{\mathbf{x}}$ such that $\|A\tilde{\mathbf{x}} - \mathbf{b}\| \leq \varepsilon$.

Up to a change in $\varepsilon$, the two notions are equivalent. Indeed, we have the chain of inequalities

$$\|A\mathbf{x} - \mathbf{b}\| \leq \|\mathbf{x} - A^{-1}\mathbf{b}\| \leq \kappa\|A\mathbf{x} - \mathbf{b}\|. \tag{1}$$

We will focus on algorithms that achieve a polylogarithmic dependence in $\varepsilon$. Prior work [6, 8, 13] focused on the first notion of approximation, which is equivalent to the second notion up to polylogarithmic factors in the complexity. Here, we focus on the second notion of approximation, for which optimal solutions can be given analytically. Indeed, in Lemma 10, we construct the optimal degree-$t$ polynomial for approximation in the second notion (see Definition 13). In Section 5, we show (numerically) that our polynomials also improve over prior work with respect to approximation in the first notion.

*From matrices to scalars.* Given a polynomial $p(x) = \sum_{t=1}^{T} c_t x^t$ with coefficients $c_t \in \mathbb{C}$, and a Hermitian matrix $A$, we define $p(A) = \sum_{t=1}^{T} c_t A^t$. If we let $A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^*$ be the eigendecomposition of $A$, then $p(A) = \sum_{i=1}^{n} p(\lambda_i)\mathbf{u}_i\mathbf{u}_i^*$.

We focus on methods to obtain a vector $\tilde{\mathbf{x}}$ that approximates $A^{-1}\mathbf{b}$ that are based on polynomials that approximate the inverse function $\lambda \mapsto \lambda^{-1}$ on the domain $[1/\kappa, 1]$ (in the case of positive definite matrices) or $D_\kappa$ (in the general case). For example, let $A$ be a Hermitian matrix with eigenvalues in $[1/\kappa, 1]$ and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial such that $|p(\lambda) - \lambda^{-1}| \leq \varepsilon$ for $\lambda \in [1/\kappa, 1]$. Then, $\tilde{\mathbf{x}} := p(A)\mathbf{b}$ satisfies

$$\|\tilde{\mathbf{x}} - A^{-1}\mathbf{b}\| = \|\sum_i (p(\lambda_i) - \lambda_i^{-1})\mathbf{u}_i\mathbf{u}_i^*\mathbf{b}\| \leq \|\sum_i (p(\lambda_i) - \lambda_i^{-1})\mathbf{u}_i\mathbf{u}_i^*\|\|\mathbf{b}\| \leq \varepsilon\|\mathbf{b}\|.$$

*Chebyshev decomposition.* It is also useful to consider the *Chebyshev decomposition* of $p(x)$, i.e., the decomposition

$$p(x) = \sum_{i=0}^{t} c_i \mathcal{T}_i(x)$$

in the basis $\{\mathcal{T}_0(x), \mathcal{T}_1(x), \ldots, \mathcal{T}_t(x)\}$, for some vector $\mathbf{c} = (c_i)_{i\in\{0,\ldots,t\}}$ of coefficients. One can give an analytic expression for the coefficients $c_i$ using the fact that the Chebyshev polynomials are orthogonal with respect to the *Chebyshev measure*, which is defined in terms of the Lebesgue measure as $d\mu(x) = (1 - x^2)^{-1/2}\,dx$. In other words, $c_i = \int_{-1}^{1} \frac{p(x)\mathcal{T}_i(x)}{\sqrt{1-x^2}}\,dx$. Note that in practice this integral is rarely computed explicitly, as there exist efficient interpolation-based methods for computing the coefficient-vector $\mathbf{c}$ [12].

## 2.2 Approximating the Inverse on $[1/\kappa, 1]$

As mentioned above, approximating the solution of a linear system $A\mathbf{x} = \mathbf{b}$ amounts to approximating the function $1/x$ on a domain that contains the spectrum of $A$. We start by assuming that $A$ is positive-definite, i.e., that all of its eigenvalues lie in the interval $[1/\kappa, 1]$. A natural polynomial to consider is the degree-$(t-1)$ Taylor expansion of $1/x$ around the point $x = 1$:

$$p_t^+(x) := \sum_{k=0}^{t-1}(1-x)^k = \frac{1-(1-x)^t}{x}.$$

Its error on the interval $[1/\kappa, 1]$ is straightforward to analyze.

Lemma 1. *We have $|xp_t^+(x) - 1| \leq \varepsilon$ for all $x \in [1/\kappa, 1]$ whenever $t \geq \kappa\log(1/\varepsilon)$.*

Proof. For all $x \in [1/\kappa, 1]$, we have

$$|xp_t^+(x) - 1| = |1 - x|^t \leq (1 - 1/\kappa)^t \leq e^{-\log(1/\varepsilon)} = \varepsilon. \qquad \square$$

Using the same reasoning as in Equation (1), we can analyze the error in the other notion of approximation.

Corollary 2. *We have* $|p_t^+(x) - 1/x| \leq \varepsilon$ *for all* $x \in [1/\kappa, 1]$ *whenever* $t \geq \kappa \log(\kappa/\varepsilon)$.

Interestingly, it turns out (see, e.g., Reference [26]) that this is exactly the polynomial that arises from the (standard, unaccelerated) gradient descent for minimizing the quadratic $\frac{1}{2}\mathbf{x}^* A\mathbf{x} - \mathbf{b}^*\mathbf{x}$ for a positive-definite matrix $A$, starting from the point $\mathbf{x}_1 = \mathbf{b}$. Given the existence of *accelerated* gradient descent methods that converge in $\widetilde{O}(\sqrt{\kappa})$ iterations, it is reasonable to expect that a corresponding polynomial with degree $\widetilde{O}(\sqrt{\kappa})$ exists as well.

Indeed, instead of $p_t^+$, one can ask what is the *best* degree-$(t-1)$ polynomial $q_t^+$? In other words, what is the degree-$(t-1)$ polynomial $q_t^+$ that minimizes

$$\max_{x \in [1/\kappa, 1]} |xq_t^+(x) - 1|. \tag{2}$$

First, observe that all such polynomials can be expressed in the form $q_t^+(x) = \frac{1 - r_t^+(x)}{x}$ where $r_t^+$ is a degree-$t$ polynomial that satisfies $r_t^+(0) = 1$.[3] Thus, our goal is to find a degree-$t$ polynomial $r_t^+(x)$ that has the smallest absolute value on the interval $[1/\kappa, 1]$ and satisfies the normalization constraint $r_t^+(0) = 1$. It turns out that we can use extremal properties of the Chebyshev polynomials $\mathcal{T}_t(x)$ to determine an optimal $r_t^+(x)$. We use the following well-known result (cf. Reference [29, Prop. 2.4]):

Lemma 3. *For any degree-$t$ polynomial $p(x)$ such that $|p(x)| \leq 1$ for all $x \in [-1, 1]$, and any $y$ such that $|y| > 1$, we have $|p(y)| \leq |\mathcal{T}_t(y)|$.*

Using the affine transformation $x \mapsto \frac{1+1/\kappa-2x}{1-1/\kappa}$ this gives the following corollary:

Corollary 4. *Let $\kappa > 1$ be real, and let $t > 0$ be an integer. Then, the polynomial*

$$r_t^+(x) = \mathcal{T}_t\left(\frac{1 + 1/\kappa - 2x}{1 - 1/\kappa}\right) \Big/ \mathcal{T}_t\left(\frac{1 + 1/\kappa}{1 - 1/\kappa}\right)$$

*is a degree-$t$ polynomial that satisfies $r_t^+(0) = 1$ and minimizes the quantity* $\max_{x \in [1/\kappa, 1]} |r_t^+(x)|$.

Note that the polynomials $r_t^+$ satisfy a Chebyshev-like 3-term recurrence. As a consequence, the polynomials $q_t^+(x) = (1 - r_t^+(x))/x$ also satisfy such a recurrence. The corresponding iterative method is known as the Chebyshev iteration.

*Remark 5 (Chebyshev Iteration).* The polynomial $q_t^+(x)$ satisfies the recurrence

$$q_{t+1}^+(x) = 2\frac{\mathcal{T}_t(\gamma)}{\mathcal{T}_{t+1}(\gamma)} \frac{\kappa + 1 - 2\kappa x}{\kappa - 1} q_t^+(x) - \frac{\mathcal{T}_{t-1}(\gamma)}{\mathcal{T}_{t+1}(\gamma)} q_{t-1}^+(x) - \frac{4\kappa}{\kappa - 1} \frac{\mathcal{T}_t(\gamma)}{\mathcal{T}_{t+1}(\gamma)}, \tag{3}$$

where $\gamma = \frac{1+1/\kappa}{1-1/\kappa}$. This recurrence corresponds to the iterative method $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{x}_1 = \mathbf{b}$ and

$$\mathbf{x}_{t+1} = 2\frac{\mathcal{T}_t(\gamma)}{\mathcal{T}_{t+1}(\gamma)} \frac{(\kappa + 1)I - 2\kappa A}{\kappa - 1} \mathbf{x}_t - \frac{\mathcal{T}_{t-1}(\gamma)}{\mathcal{T}_{t+1}(\gamma)} \mathbf{x}_{t-1} - \frac{4\kappa}{\kappa - 1} \frac{\mathcal{T}_t(\gamma)}{\mathcal{T}_{t+1}(\gamma)} \mathbf{b}.$$

Indeed, one has $\mathbf{x}_t = q_t^+(A)\mathbf{b}$.

[3] For example, in the case of the Taylor expansion, we have $r_t^+(x) = (1 - x)^t$.

The convergence rate of this method is summarized by the following theorem:

Theorem 6. *Let $\kappa > 1$ and $\varepsilon > 0$. Then, for $t \geq \frac{1}{2}\sqrt{\kappa}\log(2/\varepsilon)$, we have*

$$\left|xq_t^+(x) - 1\right| \leq \varepsilon \text{ for all } x \in [1/\kappa, 1].$$

Proof. First, we define $s(x) = \frac{1+1/\kappa-2x}{1-1/\kappa}$, so we have $r_t^+(x) = \mathcal{T}_t(s(x))/\mathcal{T}_t(s(0))$. Thus, for all $x \in [1/\kappa, 1]$, we have

$$|xq_t^+(x) - 1| = |r_t^+(x)| = \left|\mathcal{T}_t(s(x))/\mathcal{T}_t(s(0))\right|.$$

Additionally, since $|s(x)| \leq 1$ on this interval, we also have $|\mathcal{T}_t(s(x))| \leq 1$. Thus, it suffices to find $t$ for which $\mathcal{T}_t(s(0)) = \mathcal{T}_t(1 + \frac{2}{\kappa-1}) \geq \frac{1}{\varepsilon}$. Since the Chebyshev polynomial $\mathcal{T}_t(\cdot)$ can be computed as

$$\mathcal{T}_t(x) = \frac{1}{2}\left(\left(x - \sqrt{x^2-1}\right)^t + \left(x + \sqrt{x^2-1}\right)^t\right) \text{ for } |x| \geq 1, \tag{4}$$

we can conclude that $\mathcal{T}_t(s(0)) = \frac{1}{2}\left(\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^t\right) \geq \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^t$. Using the inequality $(1 + \frac{x}{n})^{n+x/2} \geq e^x$ for $x, n \geq 0$, after substituting $t = \frac{1}{2}\sqrt{\kappa}\log(2/\varepsilon)$, $x = 2$, $n = \sqrt{\kappa} - 1$, we have

$$\begin{aligned}\mathcal{T}_t(s(0)) &\geq \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^t = \frac{1}{2}\left(1 + \frac{2}{\sqrt{\kappa}-1}\right)^{(\sqrt{\kappa}-1+2/2)\frac{\log(2/\varepsilon)}{2}} \\ &\geq \frac{1}{2}\exp(\log(2/\varepsilon)) = \frac{1}{\varepsilon}.\end{aligned}$$

□

Just like in Corollary 2, we can bound the error $|q_t^+(x) - 1/x|$.

Corollary 7. *Let $\kappa > 1$ and $\varepsilon > 0$. Then, for $t \geq \frac{1}{2}\sqrt{\kappa}\log(2\kappa/\varepsilon)$, we have*

$$\left|q_t^+(x) - 1/x\right| \leq \varepsilon \text{ for all } x \in [1/\kappa, 1].$$

### 2.3 The General Case

We now return to the setting where $A$ is a Hermitian matrix and has eigenvalues in the domain $D_\kappa = [-1, -1/\kappa] \cup [1/\kappa, 1]$. In this case, instead of the (indefinite) system $A\mathbf{x} = \mathbf{b}$, we consider the equivalent positive-definite linear system $A^*A\mathbf{x} = A^*\mathbf{b}$. In other words, we approximate the inverse on $D_\kappa$ using the positive case and a simple substitution:

Corollary 8. *Let $\varepsilon > 0$, $\kappa > 1$, and let $P_t$ be any degree-$(t-1)$ polynomial such that $|yP_t(y)-1| \leq \varepsilon$ for all $y \in [1/\kappa^2, 1]$. Then, $|x^2P_t(x^2) - 1| \leq \varepsilon$ for all $x \in D_\kappa$.*

We define the following two polynomials as the respective analogs of $p_t^+$ and $q_t^+$ for $D_\kappa$:

$$p_t(x) = xp_t^+(x^2) = \frac{1 - (1-x^2)^t}{x}, \text{ and} \tag{5}$$

$$q_t(x) = xq_t^+(x^2) = \frac{1 - \mathcal{T}_t(\frac{1+1/\kappa^2-2x^2}{1-1/\kappa^2})/\mathcal{T}_t(\frac{1+\kappa^2}{1-\kappa^2})}{x}. \tag{6}$$

We call $q_t$ the *Chebyshev iteration polynomial*. Both $p_t$ and $q_t$ are degree-$(2t-1)$ polynomials, but different values of $t$ are required to achieve an $\varepsilon$-approximation of $1/x$ on $D_\kappa$. In particular, the following degrees are required:

Corollary 9. *Let $\kappa > 1$ and $\varepsilon > 0$. Then,*

*(1) $|p_t(x) - 1/x| \leq \varepsilon$ for all $x \in D_\kappa$ whenever $t \geq \kappa^2\log(\kappa/\varepsilon)$,*
*(2) $|q_t(x) - 1/x| \leq \varepsilon$ for all $x \in D_\kappa$ whenever $t \geq \frac{1}{2}\kappa\log(2\kappa/\varepsilon)$.*

Lemma 10. *Let $t \in \mathbb{N}$ and $\kappa > 1$. The polynomial $q_t$ is a degree-$(2t-1)$ polynomial that minimizes the quantity $\max_{x \in D_\kappa} |xP(x) - 1|$ among all degree-$(2t-1)$ polynomials $P \in \mathbb{R}[x]$.*

Proof. For a given $t$, we define

$$\varepsilon^+ := \min_{\substack{P^+ \in \mathbb{R}[y] \\ \deg P^+ = t-1}} \max_{y \in [1/\kappa^2, 1]} |yP^+(y) - 1|, \qquad \varepsilon := \min_{\substack{P \in \mathbb{R}[x] \\ \deg P = 2t-1}} \max_{x \in D_\kappa} |xP(x) - 1|.$$

We first show that $q_t$ certifies that $\varepsilon \le \varepsilon^+$, and then we show $\varepsilon = \varepsilon^+$. From Corollary 4, we know that $\varepsilon^+$ is achieved by the degree-$(t-1)$ polynomial $q_t^+(x) := \frac{1 - \mathcal{T}_t(s_+(x))/\mathcal{T}_t(s(0))}{x}$, where $s_+(x) := \frac{1+1/\kappa^2 - 2x}{1-1/\kappa^2}$. Then, for $q_t(x) := \frac{1-\mathcal{T}_t(s_+(x^2))/\mathcal{T}_t(s_+(0))}{x}$, we have

$$\max_{x \in D_\kappa} |xq_t(x) - 1| = \max_{x \in D_\kappa} |x^2 q_t^+(x^2) - 1| = \max_{y \in [1/\kappa^2, 1]} |yq_t^+(y) - 1| = \varepsilon^+,$$

where in the first equality we use Equation (6). We now show that $\varepsilon = \varepsilon^+$. Let $P(x)$ be a degree-$(2t-1)$ polynomial that satisfies $\max_{x \in D_\kappa} |xP(x) - 1| = \varepsilon$. We first show that $P$ is odd. To do this, decompose $P$ as $P(x) = P_{\text{even}}(x) + P_{\text{odd}}(x)$ where $P_{\text{even}}$ is even and $P_{\text{odd}}$ is odd. Then,

$$\begin{aligned}
\max_{x \in D_\kappa} |xP(x) - 1| &= \max_{x \in [1/\kappa, 1]} \max\{|xP(x) - 1|, |-xP(-x) - 1|\} \\
&= \max_{x \in [1/\kappa, 1]} \max\{|xP_{\text{odd}}(x) + xP_{\text{even}}(x) - 1|, |xP_{\text{odd}}(x) - xP_{\text{even}}(x) - 1|\} \\
&\ge \max_{x \in [1/\kappa, 1]} |xP_{\text{odd}}(x) - 1| = \max_{x \in D_\kappa} |xP_{\text{odd}}(x) - 1|.
\end{aligned}$$

Hence, replacing $P$ by $P_{\text{odd}}$ decreases $\varepsilon$, so we may assume that $P(x)$ is odd. Then, $P(x)/x$ is a degree-$(2t-2)$ even polynomial. Let $P^+(y)$ be the degree-$(t-1)$ polynomial for which $P(x)/x = P^+(x^2)$. Then, we have

$$\max_{y \in [1/\kappa^2, 1]} |yP^+(y) - 1| = \max_{x \in [1/\kappa, 1]} |x^2 P^+(x^2) - 1| = \max_{x \in D_\kappa} |xP(x) - 1| = \varepsilon.$$

This shows that $\varepsilon^+ \le \varepsilon$, which concludes the proof: $q_t$ is the degree-$(2t-1)$ polynomial that minimizes $\max_{x \in D_\kappa} |xP(x) - 1|$ over polynomials of degree $2t-1$. □

We conclude this section with a short discussion of the approach taken by CKS [8], the previous best polynomial-based QLS algorithm. The key insight of Reference [8] is that one can truncate the higher order terms of $p_t$ in the Chebyshev basis without a significant impact on the approximation error. The polynomial $p_t$ can be written in the Chebyshev basis as follows:

$$p_t(x) = 4 \sum_{j=0}^{t-1} (-1)^j \left( \frac{\sum_{i=j+1}^{t} \binom{2t}{t+i}}{2^{2t}} \right) \mathcal{T}_{2j+1}(x). \tag{7}$$

This expansion can be truncated at $\tilde{O}(\kappa)$ terms, since the Chebyshev coefficients decay exponentially. This can be shown by relating the absolute value of the $j$th coefficient (for $j = 0, 1, \ldots$) to the probability of more than $t + j$ heads appearing in $2t$ tosses of a fair coin. This probability decreases as $e^{-j^2/t}$, which can be seen by applying the Chernoff bound. Thus, starting from $p_t$, an $\varepsilon$-approximation of the inverse, we obtain an $\varepsilon$-approximation of $p_t$ by truncating the summation at $j = \sqrt{t \log(4t/\varepsilon)} = \tilde{O}(\kappa)$. For these parameters, we obtain a $2\varepsilon$-approximation of the inverse on $D_\kappa$. We refer to this polynomial as the *CKS polynomial* and we state its main property below.

Theorem 11 ([8]). *Let $\kappa > 1$, $\varepsilon > 0$ and let $\tilde{p}_t$ be the truncated degree-$(2t-1)$ CKS polynomial. Then, $|\tilde{p}_t(x) - 1/x| \le \varepsilon$ for all $x \in D_\kappa$ whenever $t \ge 1 + \sqrt{t' \log(8t'/\varepsilon)}$, where $t' = \kappa^2 \log(2\kappa/\varepsilon)$.*

## 3 QUANTUM ALGORITHMS FOR POLYNOMIAL TRANSFORMATIONS

There exist different input models that one might consider when solving the linear system problem. In the standard case of a dense matrix $A$, one might assume that all entries of $A$ are already stored in memory. Alternatively, if $A$ is sparse, then sometimes it is more efficient to consider oracle access to its nonzero entries. In the quantum setting, this *sparse-access* model is particularly amenable to speedups. In the sparse-access model, we assume that access to $A$ is provided through two oracles

$$O_{\mathrm{nz}} : |j, \ell\rangle \mapsto |j, \nu(j, \ell)\rangle \text{ and } O_A : |j, k, z\rangle \mapsto |j, k, z \oplus A_{jk}\rangle ,$$

where $\nu(j, \ell)$ is the row index of the $\ell$th nonzero entry of the $j$th column. Many quantum algorithms can be phrased naturally in terms of a different input model called the *block-encoding* model [6, 21]. (One can efficiently construct a block-encoding, given sparse access.)

*Definition 12 (Block Encoding).* Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix, and let $N \in \mathbb{N}$ be such that $n = 2^N$, and let $\mu \geq 1$. The $(N + a)$-qubit operator $U_A$ is a $(\mu, a)$-block-encoding of $A$ if it satisfies $A = \mu(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I)$.

For convenience, if we are not interested in the number of ancillary qubits $a$, then we simply call $U_A$ a $\mu$-block-encoding. In what follows, we assume that we have access to $U_A$, an (exact[4]) $(1, a)$-block-encoding of $A$. The case of $\mu$-block-encodings with $\mu > 1$ can be reduced to the former by replacing our starting matrix with $A/\mu$, which has eigenvalues in $D_{\mu\kappa}$. Furthermore, we assume that $A$ is invertible, with eigenvalues in $D_\kappa$. Finally, we assume that we have access to $U_{\mathbf{b}}$, a unitary that (exactly) prepares the state $|\mathbf{b}\rangle = \mathbf{b}/\|\mathbf{b}\|$ on input $|\mathbf{0}\rangle$: $U_{\mathbf{b}} |\mathbf{0}\rangle = |\mathbf{b}\rangle$.

We define the **quantum linear system problem (QLSP)** as follows:

*Definition 13 (Quantum Linear Systems).* Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues in $D_\kappa$, let $\mathbf{b} \in \mathbb{C}^n$, and let $\varepsilon > 0$. Given a block-encoding $U_A$ of $A$ and a state preparation oracle $U_{\mathbf{b}}$, output a state

$$|\phi\rangle = \alpha\, |0\rangle\, |\mathbf{x}\rangle + \beta\, |1\rangle\, |\psi\rangle ,$$

where $\|\, |A\mathbf{x}\rangle - |b\rangle\, \| \leq \varepsilon$, $|\psi\rangle$ is an arbitrary state, and $\alpha, \beta \in \mathbb{C}$ are such that $|\alpha|^2 + |\beta|^2 = 1$ and $|\alpha|^2 \geq 2/3$.

As mentioned before, the widely used definition from the the literature [6, 8, 13] is equivalent to Definition 13 up to a change in $\varepsilon$. In this article, we use Definition 13, as our algorithm is optimal in this sense. In Section 5, we (numerically) show that our algorithm also improves over prior work with respect to the more widely used definition.

Recent approaches for solving the QLS problem are based on applying a block-encoding of $p(A)$ to $|\mathbf{b}\rangle$. In the next two sections, we describe two ways of computing a block-encoding of $p(A)$: through the QSVT framework or by decomposing $p$ in the Chebyshev basis, computing each term individually, and combining the results using the linear combination of unitaries lemma (the LCU approach).

### 3.1 QSVT Approach

The state-of-the-art way for evaluating a polynomial quantumly is through the quantum singular value transformation framework [13]. Using QSVT, one can directly evaluate any polynomial $p$ as long as its sup-norm is suitably bounded. Here, the sup-norm of $p$ is defined as

$$\|p\|_\infty := \max_{x\in[-1,1]} |p(x)|.$$

[4]Constructing exact block-encodings of arbitrary matrices $A$ that are given in the sparse-access input model is *a priori* not possible with a finite gate set. Instead, one can construct a block-encoding of an approximation $\tilde{A}$ by allowing an overhead in the circuit depth that is proportional to $\log(\|A - \tilde{A}\|)$.

This is achieved by performing a series of rotations by angles $\Phi = (\phi_1, \ldots, \phi_t)$ on a single qubit, which induces a degree-$t$ polynomial transformation of the singular values of $A$. Determining these angles efficiently in a numerically stable way is the subject of ongoing research [7, 10, 14]. Below, we state a version of QSVT suitable for evaluating even and odd polynomials, since this is the case we are most interested in.

Theorem 14 ([13, Corollary 18], for block-encodings). *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian, and let $U_A$ be a 1-block-encoding of $A$. Let $\Pi = (|0\rangle\langle 0|)^{\otimes a} \otimes I$, and suppose that $p \in \mathbb{R}[x]$ is a degree-$t$ polynomial of parity-($t$ mod 2) satisfying $\|p\|_\infty \leq 1$. Then, there exists a $\Phi \in \mathbb{R}^t$ such that*

$$p(A) = \left(\langle +| \otimes \Pi\right)\left(|0\rangle\langle 0| \otimes U_\Phi + |1\rangle\langle 1| \otimes U_{-\Phi}\right)\left(|+\rangle \otimes \Pi\right),$$

*where $U_\Phi$ is defined as the phased alternating sequence*

$$U_\Phi := \begin{cases} e^{\mathbf{i}\phi_1(2\Pi - I)} U_A \prod_{j=1}^{(t-1)/2} \left(e^{\mathbf{i}\phi_{2j}(2\Pi-I)} U_A^* e^{\mathbf{i}\phi_{2j+1}(2\Pi-I)} U_A\right) & \text{if } n \text{ is odd, and} \\ \prod_{j=1}^{t/2} \left(e^{\mathbf{i}\phi_{2j-1}(2\Pi-I)} U_A^* e^{\mathbf{i}\phi_{2j}(2\Pi-I)} U_A\right) & \text{if } n \text{ is even.} \end{cases}$$

Note that QSVT is fundamentally limited to evaluating polynomials that are bounded by 1 in absolute value on $[-1, 1]$ (since the output is a unitary matrix). Approximations $p$ of $x^{-1}$ on $D_\kappa$ are inherently not bounded by 1 on the interval $[-1, 1]$: They are around $\kappa$ for $x = 1/\kappa$. The QSVT framework allows us to evaluate $p(x)/M$ on $A$ where $M$ is an upper bound on $\|p\|_\infty$. This subnormalization reduces the success probability of, for example, a QSVT-based QLS-solver. It is thus important to obtain polynomial approximations $p$ that moreover permit a good bound $M$.

### 3.2 LCU Approach

An alternative, prior, approach is based on the **Linear Combinations of Unitaries (LCU)** lemma [5]. It uses the fact that Chebyshev polynomials have a particularly nice vector of angles, which permits an efficient implementation of the LCU circuit.

Lemma 15 ([13, Lemma 9]). *Let $\Phi \in \mathbb{R}^t$ be such that $\phi_1 = (1-t)\frac{\pi}{2}$ and $\phi_i = \frac{\pi}{2}$ for $2 \leq i \leq t$. For this choice of $\Phi$, the polynomial $p$ from Theorem 14 is $\mathcal{T}_t$, the $t$th Chebyshev polynomial of the first kind.*

*Computing a single Chebyshev polynomial.* We consider in more detail the above circuit for computing $\mathcal{T}_{2t+1}(A)$ for a matrix $A$ with a 1-block-encoding $U_A$. Let $\Pi = |0\rangle\langle 0| \otimes I$ be the same projector as in Theorem 14 (we drop the exponent $\otimes a$ for convenience, or equivalently, we assume that the block-encoding $U_A$ has a single auxiliary qubit). By Lemma 15, the unitary

$$U_{2t+1} = e^{-\pi \mathbf{i} t(2\Pi - I)} U_A \prod_{j=1}^{t} \left(e^{\mathbf{i}\frac{\pi}{2}(2\Pi-I)} U_A^* e^{\mathbf{i}\frac{\pi}{2}(2\Pi-I)} U_A\right)$$

satisfies $(\langle 0| \otimes I) U_{2t+1} (|0\rangle \otimes I) = \mathcal{T}_{2t+1}(A)$. We first simplify the above. Note that $2\Pi - I$ has eigenvalues $\pm 1$ and therefore $e^{-\pi \mathbf{i} t(2\Pi-I)} = (-1)^t I$ and $e^{\mathbf{i}\frac{\pi}{2}(2\Pi-I)} = \mathbf{i}(2\Pi - I)$. This means that

$$\begin{aligned} U_{2t+1} &= (-1)^t U_A \prod_{j=1}^{t} \left(\mathbf{i}(2\Pi - I) U_A^* \mathbf{i}(2\Pi - I) U_A\right) \\ &= (-1)^t (\mathbf{i})^{2t} U_A \prod_{j=1}^{t} \left((2\Pi - I) U_A^* (2\Pi - I) U_A\right) \\ &= U_A \prod_{j=1}^{t} \Big(\underbrace{(2\Pi - I) U_A^* (2\Pi - I) U_A}_{=:W}\Big) = U_A W^t. \end{aligned}$$

In other words, $U_{2t+1}$ can be viewed as $t$ applications of the unitary $W$, followed by a single application of $U_A$, see also Reference [8, Lemma 16]. The circuit for even Chebyshev polynomials $U_{2t}$ is very similar and can be obtained from $U_{2t+1}$ by removing the final application of (left multiplication by) $U_A$—however, since we are ultimately interested in implementing the inverse, an odd function, we do not describe the circuit in more detail.

*Computing a linear combination of Chebyshev polynomials.* Given the above circuit that computes block-encodings of $\mathcal{T}_{2k+1}(A)$ for $k \geq 0$, the next step is to compute a block-encoding of linear combinations of the form

$$p(A) = \sum_{i=0}^{t-1} c_i \mathcal{T}_{2i+1}(A). \tag{8}$$

This can be achieved using a version of the LCU algorithm due to Reference [8]. In particular, the key to an efficient implementation of the linear combination $\sum_{i=0}^{t-1} c_i U_{2i+1}$ is the efficient implementation of the operator $\sum_{i=0}^{t-1} |i\rangle\langle i| \otimes U_{2i+1}$, which we achieve by introducing an $l = (\lceil \log_2 t \rceil + 1)$-qubit *counter* register and successively applying $W, W^2, W^4, \ldots, W^{2^{l-1}}$ controlled on qubits $0, 1, \ldots, l-1$ of the counter, followed by a single application of $U_A$ at the end. In Reference [8, Theorem 4] this circuit is analyzed for a specific polynomial-approximation of the inverse. The analysis naturally extends to arbitrary polynomials of the form (8).

Theorem 16 (based on Reference [8]). *Let $A$ be a Hermitian matrix with eigenvalues in $D_\kappa$, let $U_A$ be its block-encoding, and let $U_{\sqrt{\mathbf{c}}}$ be a unitary that prepares the state $\frac{1}{\sqrt{\|\mathbf{c}\|_1}} \sum_{i=0}^{t-1} \sqrt{c_i}\, |i\rangle$. Then, there exists an algorithm that computes a $\|\mathbf{c}\|_1$-block-encoding of $p(A)$ using $t+1$ calls to controlled versions of $U_A$ and $U_A^*$, and a single call to each of $U_{\sqrt{\mathbf{c}}}$ and $U_{\sqrt{\mathbf{c}}}^*$. This circuit uses a logarithmic number of additional qubits and has a gate complexity of $O(t \operatorname{polylog}(nt\kappa/\varepsilon))$.*

Compared to the QSVT approach, for this circuit, we only need to compute the Chebyshev coefficients $\mathbf{c}$, as opposed to the vector of angles $\mathbf{\Phi}$—this comes, however, at the cost of using $O(\log t)$ additional qubits. Moreover, the coefficient 1-norm $\|\mathbf{c}\|_1$ represents an upper bound for $\|p\|_\infty$, since

$$|p(x)| = \left| \sum_{i=0}^{t} c_i \mathcal{T}_i(x) \right| \leq \sum_{i=0}^{t} |c_i| \cdot |\mathcal{T}_i(x)| \leq \|\mathbf{c}\|_1, \quad \text{for } |x| \leq 1. \tag{9}$$

A natural question is how tight this bound is for general degree-$t$ polynomials $p$ with $\|p\|_\infty \leq 1$. By norm conversion (Equation (14), in particular), the ratio $\|\mathbf{c}\|_1 / \|p\|_\infty$ is provably upper bounded by $O(\sqrt{t})$ but in Appendix A, we observe that for many "interesting" functions the ratio $\|\mathbf{c}\|_1 / \|p\|_\infty$ is in fact only $O(\log(t))$. A notable exception is the complex exponential $e^{\mathrm{i}\kappa x}$ (and thus $\sin(\kappa x)$ and $\cos(\kappa x)$) for which numerical experiments suggest that it attains the $O(\sqrt{t})$ upper bound. In Appendix A.3, we show how to overcome this limitation by composing easily implementable functions.

## 4 A QLS-ALGORITHM BASED ON $Q_T$

As mentioned before, our main result is the following explicit upper bound on the Chebyshev coefficient 1-norm of $q_t$:

Theorem 17 (Main Result). *For all $t \in \mathbb{N}$, the Chebyshev coefficient 1-norm of $q_t$ is bounded as $\|\mathbf{c}_t\|_1 \leq 2(1 + \frac{1}{\mathcal{T}_t(s(0))})t$. In particular, for $t \geq \frac{1}{2}\kappa \log(2\kappa^2/\varepsilon)$, we have $\|\mathbf{c}_t\|_1 \leq 2(1 + \varepsilon/\kappa^2)t$.*

The proof of this Theorem is split between Section 4.1 and Lemma 19. As a corollary, we get a QSVT-based QLS algorithm that can be described as applying the polynomial $q_t$ to a

1-block-encoding of the input matrix $A$. This yields an $O(t)$-block-encoding of $q_t(A)$, which can then be applied to the input state $|\mathbf{b}\rangle$. Formally, we show the following:

Corollary 18 (QSVT-based Algorithm). *Let $A$ be a Hermitian matrix with eigenvalues in $D_\kappa$, let $U_A$ be a 1-block-encoding of $A$, and let $\varepsilon > 0$. Then, for $t \geq \frac{1}{2}\kappa \log(2\kappa^2/\varepsilon)$, a $2(1+\varepsilon/\kappa^2)t$-block-encoding of $q_t(A)$ can be constructed using $2t-1$ calls to $U_A$ and $U_A^*$.*

Proof. The algorithm consists of applying QSVT (Theorem 14) to the polynomial $q_t(x)/\left\|q_t\right\|_\infty$. This allows us to construct a $\left\|q_t\right\|_\infty$-block-encoding of $q_t(A)$ with the desired complexity. It remains to upper bound $\left\|q_t\right\|_\infty$ by $2(1+\varepsilon/\kappa^2)t$. Motivated by Equation (9), it suffices to upper bound the 1-norm of the vector $\mathbf{c}$ of coefficients of $q_t$ in the Chebyshev basis (again by $2(1+\varepsilon/\kappa^2)t$)—this is guaranteed by Theorem 17. □

The block-encoding of $q_t(A)$ can now be used as a black-box replacement for the block-encoding of the corresponding CKS polynomial evaluated at $A$. For example, using variable-time amplitude amplification, an $\widetilde{O}(\kappa)$-query (to $U_A$) complexity QLS algorithm can be derived. We refer the reader to References [8, 13, 22] for an overview of these techniques.

As an alternative approach, one could use the fact that $\|\mathbf{c}_t\|_1$ is bounded to evaluate $q_t$ via LCU (Theorem 16). At the cost of using $O(\log t)$ additional qubits, an LCU-based approach would yield a more "natural" quantum algorithm, that does away with the classical angle computation preprocessing step required by QSVT—computing these angles efficiently in a numerically stable way is the subject of ongoing research [7, 10, 14].

### 4.1 Bounding the Chebyshev Coefficients

As discussed above, to apply (a normalized version of) $q_t$ to a block-encoding of a Hermitian matrix with eigenvalues in $D_\kappa$, we need a bound on the sup-norm of $q_t$ on the interval $[-1,1]$. To derive such a bound, we express $q_t$ in the basis of Chebyshev polynomials. Each of the Chebyshev polynomials has sup-norm equal to 1 and therefore a bound on the 1-norm of the coefficient vector provides a bound on the sup-norm of $q_t$. Recall that, since $q_t$ is an odd polynomial, its expansion in the Chebyshev basis only involves the odd-degree Chebyshev polynomials. That is, we can write

$$q_t(x) = \sum_{i=0}^{t-1} c_{t,i}\mathcal{T}_{2i+1}(x) \tag{10}$$

for some vector $\mathbf{c}_t = (c_{t,i})_{i\in\{0,\ldots,t-1\}}$ of coefficients. One can give an analytic expression for $c_{t,i}$ using the fact that the Chebyshev polynomials are orthogonal with respect to the *Chebyshev measure*. Here, we take a different approach and use the following discrete orthogonality relations (see, e.g., Reference [28]): Fix a degree $m \in \mathbb{N}$ and let $\{x_1,\ldots,x_m\}$ be the roots of $\mathcal{T}_m(x)$. The $x_k$'s are called the *Chebyshev nodes* and they admit an analytic formula:

$$x_k = \cos\left(\frac{(k-\frac{1}{2})\pi}{m}\right) \qquad \text{for } k = 1,\ldots,m. \tag{11}$$

The discrete orthogonality relation that we will use is the following: For $0 \leq i,j < m$, we have

$$\sum_{k=1}^{m} \mathcal{T}_i(x_k)\mathcal{T}_j(x_k) = \begin{cases} m & \text{if } i=j=0, \\ \frac{m}{2} & \text{if } i=j<m, \\ 0 & \text{if } i \neq j. \end{cases} \tag{12}$$

Since $q_t$ is a polynomial of degree $2t-1$, we will use the discrete orthogonality conditions corresponding to $m = 2t$ to recover the coefficient of $\mathcal{T}_{2i+1}$ in $q_t$. We have

$$c_{t,i} = \frac{1}{t}\sum_{k=1}^{2t} q_t(x_k)\mathcal{T}_{2i+1}(x_k) \tag{13}$$

for all $i \in \{0, 1, \ldots, t-1\}$. We can equivalently write this in matrix form, $\mathbf{c}_t = \frac{1}{t}\boldsymbol{\mathcal{T}}_t \mathbf{q}_t$, where

$$\boldsymbol{\mathcal{T}}_t = \begin{bmatrix} \mathcal{T}_1(x_1) & \mathcal{T}_1(x_2) & \ldots & \mathcal{T}_1(x_{2t}) \\ \mathcal{T}_3(x_1) & \mathcal{T}_3(x_2) & \ldots & \mathcal{T}_3(x_{2t}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{T}_{2t-1}(x_1) & \mathcal{T}_{2t-1}(x_2) & \ldots & \mathcal{T}_{2t-1}(x_{2t}) \end{bmatrix} \text{ and } \mathbf{q}_t = \begin{bmatrix} q_t(x_1) \\ q_t(x_2) \\ \vdots \\ q_t(x_{2t}) \end{bmatrix}.$$

Our goal is to show that $\|\mathbf{c}_t\|_1 \leq C \cdot t$ for a small constant $C$. To do so, we first use the Cauchy-Schwarz inequality to obtain

$$\|\mathbf{c}_t\|_1 \leq \sqrt{t}\|\mathbf{c}_t\|_2 = \frac{1}{\sqrt{t}}\|\boldsymbol{\mathcal{T}}_t\mathbf{q}_t\|_2 \leq \frac{\|\boldsymbol{\mathcal{T}}_t\|}{\sqrt{t}}\|\mathbf{q}_t\|_2 = \|\mathbf{q}_t\|_2, \tag{14}$$

where the last equality follows from the discrete orthogonality relations Equation (12): We see that $\boldsymbol{\mathcal{T}}_t\boldsymbol{\mathcal{T}}_t^* = tI_t$ and therefore $\|\boldsymbol{\mathcal{T}}_t\| = \sqrt{t}$. Thus, bounding $\|\mathbf{q}_t\|_2$ would conclude the proof of Theorem 17.

Lemma 19. *We have $\|\mathbf{q}_t\|_2 \leq 2(1 + \frac{1}{\mathcal{T}_t(s(0))})t$ for all $t \in \mathbb{N}$. In particular, for $t \geq \frac{1}{2}\kappa\log(2\kappa^2/\varepsilon)$, we have $\|\mathbf{q}_t\|_2 \leq 2(1 + \varepsilon/\kappa^2)t$.*

Proof. We start by bounding $|q_t(x)|$ on $[-1, 1]$, and we recall that

$$q_t(x) = \frac{1 - \mathcal{T}_t(s(x))/\mathcal{T}_t(s(0))}{x}, \quad \text{where} \quad s(x) = \frac{1 + 1/\kappa^2 - 2x^2}{1 - 1/\kappa^2}.$$

On one hand, when $x \in D_\kappa$, we have $s(x) \in [-1, 1]$ and thus $|1 - \mathcal{T}_t(s(x))/\mathcal{T}_t(s(0))| \leq 1 + 1/\mathcal{T}_t(s(0))$. On the other hand, when $|x| \leq 1/\kappa$, we have $1 \leq s(x) \leq s(0) = \frac{1+1/\kappa^2}{1-1/\kappa^2}$. Since $\mathcal{T}_t(x)$ is increasing for $x \geq 1$, it follows that $0 \leq 1 - \mathcal{T}_t(s(x))/\mathcal{T}_t(s(0)) \leq 1$ for all $|x| \leq 1/\kappa$. Together this shows that

$$|q_t(x)| = \left|\frac{1 - \mathcal{T}_t(s(x))/\mathcal{T}_t(s(0))}{x}\right| \leq \frac{1 + 1/\mathcal{T}_t(s(0))}{|x|} \qquad \text{for all } x \in [-1, 1] \setminus \{0\}.$$

We now bound the norm of $\mathbf{q}_t$. We have

$$\|\mathbf{q}_t\|^2 = \sum_{k=1}^{2t} q_t(x_k)^2 \leq \left(1 + \frac{1}{\mathcal{T}_t(s(0))}\right)^2 \sum_{k=1}^{2t}\frac{1}{x_k^2} = \left(1 + \frac{1}{\mathcal{T}_t(s(0))}\right)^2 \sum_{k=1}^{2t}\frac{1}{\cos^2\left(\frac{2k-1}{4t}\pi\right)},$$

where we substituted the exact expression for the Chebyshev nodes $x_k = \cos\left(\frac{2k-1}{4t}\pi\right)$. Moreover, we have

$$\cos^2\left(\frac{2(2t-k+1)-1}{4t}\pi\right) = \cos^2\left(\frac{2k-1}{4t}\pi\right) = \frac{1 - \cos\left(\frac{2k-1}{2t}\pi\right)}{2} \quad \text{for all} \quad 1 \leq k \leq t,$$

where the first equality comes from $x_{2t-k+1} = -x_k$. Therefore, we have

$$\|\mathbf{q}_t\|^2 \leq 4\left(1 + \frac{1}{\mathcal{T}_t(s(0))}\right)^2 \sum_{k=1}^{t}\frac{1}{1 - \cos(\frac{2k-1}{2t}\pi)}.$$

We note that the roots of $\mathcal{T}_t(x)$ are exactly $\cos(\frac{2k-1}{2t}\pi)$. For any polynomial $P(x) = C\prod_{k=1}^{t}(x - r_k)$, the following identity holds for all $x$ for which $P(x) \neq 0$:

$$\sum_{k=1}^{t} \frac{1}{x - r_k} = \frac{P'(x)}{P(x)}.$$

Applying the above to $P(x) = \mathcal{T}_t(x)$ and $x = 1$ (which is not a root of $\mathcal{T}_t$), we get

$$\sum_{k=1}^{t} \frac{1}{1 - \cos(\frac{2k-1}{2t}\pi)} = \frac{\mathcal{T}_t'(1)}{\mathcal{T}_t(1)} = \frac{t \cdot \mathcal{U}_{t-1}(1)}{1} = t^2.$$

This concludes the main part of the proof: We have shown that $\|\mathbf{q}_t\| \leq 2(1 + \frac{1}{\mathcal{T}_t(s(0))})t$.

Finally, for $t \geq \frac{1}{2}\kappa\log(2\kappa^2/\varepsilon)$, we bound $1/\mathcal{T}_t(s(0))$ as in the proof of Corollary 7. Namely, using the same inequalities, we have

$$\mathcal{T}_t(s(0)) \geq \frac{1}{2}\left(\frac{\kappa+1}{\kappa-1}\right)^t \geq \frac{1}{2}\left(1 + \frac{2}{\kappa-1}\right)^{\frac{1}{2}\kappa\log(2\kappa^2/\varepsilon)} \geq \frac{\kappa^2}{\varepsilon}. \qquad \square$$

Combining this lemma with Equation (14), we derive the same bound for $\|\mathbf{c}_t\|_1$, thus completing the proof of Theorem 17.

## 4.2 Efficiently Computing the Coefficients

In the case of evaluating $q_t$ via LCU, one question of practical relevance is how to compute the coefficients $\mathbf{c}_t$. Naively using the recurrence (3) to compute $\mathbf{c}_t$ gives rise to an algorithm with $O(t^2)$ arithmetic operations with real numbers. Alternatively, one can use FFT-based Chebyshev interpolation algorithms that can compute $\mathbf{c}_t$ with $O(t\log t)$ operations given the vector $\mathbf{q}_t$ of the values of $q_t(x)$ at the order-$t$ Chebyshev nodes [12]. Thus, to get an $O(t\log t)$-operation algorithm for computing $\mathbf{c}_t$, it suffices to show that $q_t(x)$ can be evaluated at a single Chebyshev node $x_k$ with $O(\log t)$-operations. Given the form of $q_t$, this means that we need to compute $\mathcal{T}_t(s(x_k))$ with $O(\log t)$ operations. One way to do this is via the degree-halving identities

$$\mathcal{T}_{2t}(x) = 2\mathcal{T}_t(x)^2 - 1 \quad \text{and} \quad \mathcal{T}_{2t+1}(x) = 2\mathcal{T}_{t+1}(x)\mathcal{T}_t(x) - x.$$

## 4.3 A More Natural Quantum Algorithm?

Given the reduction of the general linear system problem to the PD case (Corollary 8), one might be tempted to mirror this reduction when designing a quantum algorithm, with the goal of achieving $\tilde{O}(\sqrt{\kappa})$ complexity for solving PD systems. The input of such an algorithm would be a (block-encoding of a) Hermitian matrix $A$ with eigenvalues in $[1/\kappa, 1]$, and the output would be a block-encoding of $q_t^+(A)$. To evaluate this polynomial using QSVT, we first need to normalize it by dividing it by $\max_{x\in[-1,1]}|q_t^+(x)|$. It turns out that this maximum grows exponentially with $t$: One can lower bound it by $|q_t^+(-1)|$ and we have

$$\begin{aligned}|q_t^+(-1)| &\geq \frac{\mathcal{T}_t\left(\frac{1+1/\kappa+2}{1-1/\kappa}\right)}{\mathcal{T}_t\left(\frac{1+1/\kappa}{1-1/\kappa}\right)} - 1 = \frac{\mathcal{T}_t(3 + 4/(\kappa-1))}{\mathcal{T}_t(1 + 2/(\kappa-1))} - 1 \geq \frac{\mathcal{T}_t(3)}{\mathcal{T}_t(2)} - 1 \\ &= \frac{\left(3-2\sqrt{2}\right)^t + \left(3+2\sqrt{2}\right)^t}{\left(2-\sqrt{3}\right)^t + \left(2+\sqrt{2}\right)^t} - 1 \geq \frac{1}{2}\left(\frac{3+2\sqrt{2}}{2+\sqrt{3}}\right)^t - 1 \geq \frac{1}{2}\left(\frac{3}{2}\right)^t - 1,\end{aligned}$$

where we subsequently use the definition of $q_t^+$, monotonicity of $\mathcal{T}_t(x)$ for $x \geq 1$, and the explicit expression Equation (4) for $\mathcal{T}_t(x)$ for $x \geq 1$. Therefore, amplifying the output of QSVT would take exponential time. In the case of LCU, the coefficient 1-norm is lower bounded by $|q_t^+(-1)|$ (by Equation (9)), so the output of a LCU-based algorithm would also need to be amplified exponentially. Alternative approaches of multiplying $q_t^+(x)$ by a rectangle function that is close to 1 on $[1/\kappa, 1]$ and close to 0 elsewhere are similarly fruitless, as the degree of the resulting approximation polynomial would become linear in $\kappa$. It should be noted, however, that these issues can be avoided if we assume that the mapping $x \mapsto \frac{1+1/\kappa-2x}{1-1/\kappa}$ has already been performed "ahead of time": In Reference [24], Orsucci and Dunjko have shown that PD matrices can indeed be inverted in $\widetilde{O}(\sqrt{\kappa})$, provided that a block-encoding of $I - \alpha A$ is given as input (for suitable $\alpha$).

Another natural alternative approach would be to quantize a method such as momentum gradient descent, which also converges in $\widetilde{O}(\sqrt{\kappa})$ for PD matrices [26]. One way to achieve this would be using the approach of Kerenidis and Prakash [16], who quantized the basic gradient descent algorithm by implementing the recurrence $\mathbf{r}_{t+1} = (I - \eta A)\mathbf{r}_t$ satisfied by the differences $\mathbf{r}_t := \mathbf{x}_t - \mathbf{x}_{t-1}$ of successive iterates. Applying this idea to momentum gradient descent, one gets a recurrence involving two successive differences:

$$\begin{bmatrix} \mathbf{r}_{t+1} \\ \mathbf{r}_t \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)I - \eta A & -\beta I \\ I & 0 \end{bmatrix}}_{M} \begin{bmatrix} \mathbf{r}_t \\ \mathbf{r}_{t-1} \end{bmatrix},$$

for suitable choices of $\eta$ and $\beta$. For example, following Reference [26, Chapter 3], one can set $\eta = 4/(1+\sqrt{1/\kappa})^2$ and $\beta = \left(1 - 2/(1+\sqrt{\kappa})\right)^2$. Implementing a similar approach as in Reference [16] would require the construction of $O(1)$-block-encodings of powers of $M$. In particular, this would require $M$ to have a small norm. Unfortunately, for large enough $\kappa \geq 9$ and the above choice of $\eta, \beta$, one has $\|M\| \geq \sqrt{2}$, which means that a block-encoding of $M^t$ needs to have subnormalization at least $2^{t/2}$.

## 5 COMPARISON WITH PREVIOUS POLYNOMIAL-BASED QLS-SOLVERS

In Lemma 10, we saw that the Chebyshev iteration polynomial $q_t$ is the degree-$(2t-1)$ polynomial that minimizes the error

$$\max_{x \in D_\kappa} \left| xP(x) - 1 \right|$$

over all polynomials of degree $2t - 1$. This implies that the CKS polynomial attains a larger error than the Chebyshev iteration polynomial, or conversely requires a higher degree to reach the same error on $D_\kappa$. In Table 1, we use Corollary 9 and Theorem 11 to compute the degree required to achieve error $\varepsilon$ on $D_\kappa$ and observe that the degree of the CKS polynomial is roughly twice the degree of the corresponding Chebyshev iteration polynomial.

Another way of comparing these polynomials is to compute their errors according to

$$\max_{x \in D_\kappa} \left| P(x) - 1/x \right|$$

(corresponding to definition 1). The optimal polynomial (also called the *minimax* polynomial) according to this definition has no explicit description, but it can still be computed relatively efficiently using the Remez exchange algorithm—see, e.g., Reference [25] for a practical implementation. In Reference [10] the authors compared the minimax and the CKS polynomials. In Figures 1 and 2, we add the Chebyshev iteration polynomials to this comparison. In particular, we show the errors $\max_{x \in D_\kappa} \left| P(x) - 1/x \right|$ where $P$ is the minimax, CKS and Chebyshev iteration polynomial.

Table 1. Degrees of Approximation Polynomials for a Given Condition Condition Number $\kappa$ and Error $\varepsilon$, Computed According to Corollary 9

(a) CKS polynomial

| $\kappa$ \ $\varepsilon$ | 0.5 | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ |
|---|---|---|---|---|
| 2 | 15 | 33 | 53 | 71 |
| 10 | 115 | 203 | 301 | 399 |
| 100 | 1,819 | 2,687 | 3,669 | 4,633 |
| 1,000 | 24,913 | 33,515 | 43,337 | 52,989 |

(b) Chebyshev iteration

| $\kappa$ \ $\varepsilon$ | 0.5 | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ |
|---|---|---|---|---|
| 2 | 5 | 11 | 21 | 31 |
| 10 | 37 | 77 | 123 | 169 |
| 100 | 599 | 991 | 1,451 | 1,911 |
| 1,000 | 8,295 | 12,207 | 16,811 | 21,417 |

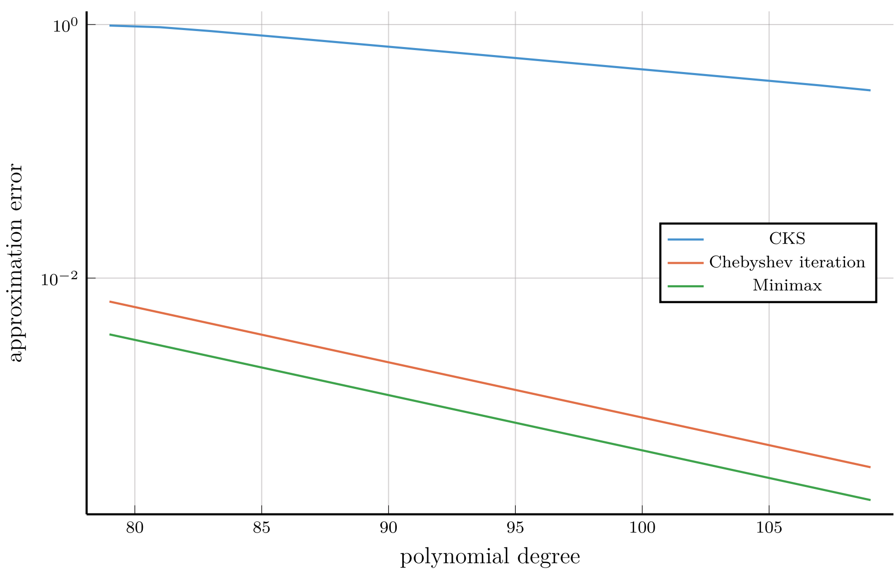


Fig. 1. Approximation error for a fixed condition number $\kappa = 10$ and varying degrees.

In Figure 1, we see that that for a fixed condition number, the convergence is linear for all polynomials, with the CKS polynomial being the slowest to converge (i.e., for the same degree, the difference in errors is a few orders of magnitude). Conversely, in Figure 2, we see that with polynomials of a fixed degree, the error of the CKS is an order of magnitude higher, no matter the condition number.

We conclude this section with a remark on the (CPU) time needed to compute (the coefficients of) these polynomials. On the authors' hardware (an Intel Xeon Silver 4110 CPU), computing all the CKS and Chebyshev iteration polynomials from Figures 1 and 2 only took a few seconds, whereas computing the corresponding minimax polynomials took several hours. One reason for this is that while the former can be computed using simple and efficient machine-precision operations, the latter requires (at least in the implementation of Reference [10]) dealing with arbitrary-precision floating-point numbers.

## 6 QUERY LOWER BOUNDS

So far, we have been considering algorithms (i.e., upper bounds) for the QLS problem. The complexity of the best algorithm for the QLS problem depends linearly on $\kappa$ (we ignore the polylogarithmic

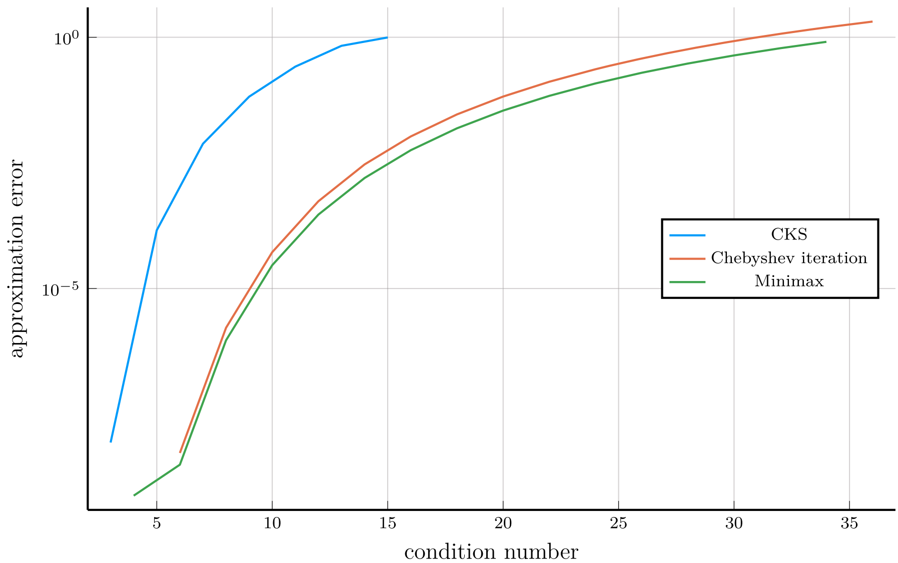


Fig. 2. Approximation error for a fixed degree $2t - 1 = 127$ and varying condition numbers.

factors in this section), so a natural question is whether this dependence is optimal. In Reference [15], it has been shown that this is indeed the case: In the sparse access input model (the setting in which such lower bounds are usually proven), the complexity of QLS for general systems is $\Omega(\min(\kappa, n))$. Recently, it has been shown [24] that the same $\Omega(\min(\kappa, n))$ lower bound even holds for the restriction of QLS to PD matrices—this is surprising, since in the classical setting a $\sqrt{\kappa}$-separation exists between the general and the PD case. We note that both of these lower bounds apply when the output of the QLS solver is the quantum state $|A^{-1}\mathbf{b}\rangle$. As a consequence, one can show that computing a classical description of $A^{-1}\mathbf{b}$ is just as hard.

Both of the above results apply to the small-$\kappa$ regime. In particular, they leave open the possibility of a $o(n^\omega)$-time quantum algorithm for solving linear systems (with classical output), where $\omega < 2.373$ is the matrix multiplication exponent. The existence of such an algorithm would speed up many classical optimization algorithms (e.g., interior point methods) in a black-box way. In Reference [11] it was shown that one cannot obtain a large quantum speedup when the output is required to be classical: $\Omega(n^2)$ quantum queries to the entries of $A$ are needed to obtain a classical description of a single coordinate of $A^{-1}e_n$, where $e_n$ is the $n$th standard basis vector in $\mathbb{R}^n$. The statement is robust in the following sense: After normalizing $A^{-1}e_n$, it requires $\Omega(n^2)$ quantum queries to the entries of $A$ even to obtain a $\delta$-additive approximation of the first coordinate for some $\delta = O(1/n^2)$. We present a simplified proof of this result of Reference [11] at the end of this section. Note that this high precision prevents one from lifting the bound to the quantum-output setting: To obtain a $\delta$-additive approximation of a single coordinate of $|A^{-1}b\rangle$, one can use roughly $1/\delta$ rounds of amplitude estimation on a QLS-solver $\mathcal{A}$. With $\delta = O(1/n^2)$ this only implies that $n^2 \cdot \mathrm{cost}(\mathcal{A}) = \Omega(n^2)$. A second type of quantum lower bound is described in Reference [13, Theorem 73]: Roughly speaking, if a (smooth) function $f : I \to [-1, 1]$ has a derivative whose absolute value is $d$, then $\Omega(d)$ uses of a 1-block-encoding $U_A$ of $A$ are needed to create a block-encoding of $f(A)$. Here, $I$ is a subset of $[-1, 1]$ that contains the eigenvalues of the Hermitian matrix $A$ and the bound on the derivative needs to be attained "away from the boundary," e.g., in the interval $[-1/2, 1/2]$. Applied to $f(x) = 1/(\kappa x)$, this shows that indeed $\Omega(\kappa)$ applications of $U_A$ are needed to create a block-encoding of $A^{-1}$. As mentioned before, a block-encoding of $A^{-1}$ can be combined with a state preparation oracle for $\mathbf{b}$ to solve the QLS problem. Such a strategy, however, naturally incurs a $\kappa$-dependence in the runtime, and it remains an interesting open question whether one could solve the QLS problem (with quantum output!) without such a dependence in $\kappa$ and in time $o(n^{\omega})$.

### 6.1 Lower Bound for Matrix Inversion with Classical Output

We present a simplified proof of a matrix-inversion lower bound result of Reference [11]. It is based on the quantum query complexity of the majority function $\mathrm{MAJ}_n : \{0,1\}^n \longrightarrow \{0,1\}$, which takes value 1 on input $\mathbf{x}$ if and only if $\sum_{i\in[n]} x_i > n/2$. It is well known that the quantum query complexity of $\mathrm{MAJ}_n$ is $\Theta(n)$ [4].

Lemma 20. *Let $X \in \{0,1\}^{n\times n}$. Then, the matrix $A \in \{0,1\}^{(2n+2)\times(2n+2)}$ defined as*

$$A = \left[\begin{array}{cc|cc} 0 & 1_n^* & 0 & 0 \\ 1_n & 0 & X & 0 \\ \hline 0 & X^* & 0 & 1_n \\ 0 & 0 & 1_n^* & 0 \end{array}\right]$$

*satisfies $(A^3)_{1,2n+2} = \sum_{i=1}^n \sum_{j=1}^n X_{i,j}$.*

Proof. $A$ is the adjacency matrix of an undirected graph that can be described as follows: We start with a bipartite graph between two sets of $n$ vertices whose edge set is described by $X$, then we add two vertices labeled 1 and $2n+2$ that we connect, respectively, to the first set of vertices and the second set of vertices. The entry $(1, 2n+2)$ of $A^3$ counts the number of paths of length 3 from 1 to $2n+2$ in this graph. This equals the number of edges between the sets $\{2,\dots,n+1\}$ and $\{n+2,\dots,2n+1\}$, that is, $(A^3)_{1,2n+2} = \sum_{i=1}^n \sum_{j=1}^n X_{i,j}$. □

Corollary 21. *Let $A \in \{0,1\}^{n\times n}$. Determining a single off-diagonal entry of $A^3$, with success probability $\geq 2/3$, takes $\Theta(n^2)$ quantum queries to $A$.*

Lemma 22. *Let $A \in \{0,1\}^{n\times n}$. Then, for $N = 4n$, the matrix $B \in \{0,1\}^{N\times N}$ defined by*

$$B = \begin{bmatrix} I & A & & \\ & I & A & \\ & & I & A \\ & & & I \end{bmatrix},$$

*satisfies $(B^{-1})_{1,N} = -(A^3)_{1,n}$.*

Proof. It is straightforward to verify that the inverse of $B$ is

$$B^{-1} = \begin{bmatrix} I & -A & A^2 & -A^3 \\ & I & -A & A^2 \\ & & I & -A \\ & & & I \end{bmatrix}.$$ □

If $A$ is the adjacency matrix of the directed version of the graph described in Lemma 20, then we can also compute the norm of the last column as follows:

$$\left\|B^{-1}\mathbf{e}_{8n+8}\right\|^2 = \left(\sum_{i,j} X_{i,j}\right)^2 + \sum_i \left(\sum_j X_{i,j}\right)^2 + n + 1.$$

In particular, for the hard instances (where $|n/2 - \sum_{i,j} X_{i,j}| \leq 1$), we have that $\left\|B^{-1}\mathbf{e}_{8n+8}\right\| = \Theta(n^2)$.

Corollary 23. *Let $A \in \{0,1\}^{n\times n}$. Determining a single off-diagonal entry of $A^{-1}$ up to precision $< 1/2$, with success probability $\geq 2/3$, takes $\Theta(n^2)$ quantum queries to $A$.*

## APPENDIX

## A EXAMPLES OF FUNCTIONS WITH BOUNDED CHEBYSHEV COEFFICIENT NORMS

The inverse function is not the only function that can be efficiently evaluated using LCU of Chebyshev polynomials. Here, we discuss several families of functions for which the 1-norm of the Chebyshev coefficients is of the order log(degree).

### A.1 Simple Examples

We first observe that the monomial $x^n$ has the following Chebyshev expansion:

$$x^n = 2^{1-n} \sum_{j=0,\ n-j \text{ even}}^{n}{}' \binom{n}{\frac{n-j}{2}} \mathcal{T}_j(x),$$

where the prime at the sum symbol indicates that the contribution of $j = 0$ needs to be halved (if it appears). The sum of these coefficients is bounded by 1. This implies that for any polynomial the 1-norm of the coefficients in the Chebyshev basis is at most the 1-norm of the coefficients in the monomial basis. This means, for example, that the Chebyshev coefficient 1-norm of the scaled exponential is at most 1. Similarly, for a degree-$n$ Taylor approximation of the (scaled) logarithm the 1-norm grows as $O(\log n)$. In particular, for the scaled exponential function $e^{\kappa(x-1)}$ that maps the interval $[-1,1]$ to $[e^{-2\kappa},1]$, we have the following Taylor expansions for $\kappa \geq 1$:

$$e^{\kappa(x-1)} = e^{-\kappa} \sum_{j=0}^{\infty} \frac{(\kappa x)^j}{j!}.$$

Clearly for any truncation, the 1-norm of the coefficients is at most 1.

Similarly, one can consider the logarithm on the interval $[1/\kappa, 1]$. We view it as a function $\mathrm{slog}_\kappa : [-1,1] \to [\log(1/\kappa), 1]$ as follows:

$$\mathrm{slog}_\kappa(x) := \log(1/\kappa + ((x+1)/2)(1-1/\kappa)) = \log\left(\frac{\kappa+1}{2\kappa}\left(1 + \frac{\kappa-1}{\kappa+1}x\right)\right).$$

Then, its Taylor expansion becomes

$$\mathrm{slog}_\kappa(x) = \log\left(\frac{\kappa+1}{2\kappa}\right) + \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j}\left(\frac{\kappa-1}{\kappa+1}\right)^j x^j.$$

For small values of $n$, the sum of the absolute values of the first $n$ coefficients scales as $\log(n)$. (However, as $n \to \infty$, the sum remains bounded by roughly $\kappa$, as can be seen, for instance, using the Chebyshev inequality.)

### A.2 Approximating Discontinuities—the Error Function

Some more interesting examples are the sign and the rectangle functions, defined as

$$\mathrm{sign}(x) := \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0, \end{cases} \quad \text{and} \quad \Pi(x) := \begin{cases} 1 & \text{if } |x| \leq 1/2, \\ 0 & \text{else.} \end{cases}$$

It is well-known [13, 20] that the error-function $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-z^2}\,\mathrm{d}z$ is a fundamental building block for approximating discontinuous functions. For example, given $\varepsilon, \delta > 0$, there exists a choice of $\kappa = O(\mathrm{polylog}(1/\varepsilon)/\delta)$ such that $\mathrm{erf}(\kappa x)$ is $\varepsilon$-close to $\mathrm{sign}(x)$ on $[-1,1]\setminus[-\delta,\delta]$. We show below that the 1-norm of the coefficients of the Chebyshev series of $\mathrm{erf}(\kappa x)$ is $O(\log\kappa)$. We start with the following expansion from Reference [20] where $I_j$ is the $j$th Bessel function of the first kind:

$$\mathrm{erf}(\kappa x) = \frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\left(I_0(\kappa^2/2)x + \sum_{j=1}^{\infty} I_j(\kappa^2/2)(-1)^j\left(\frac{\mathcal{T}_{2j+1}(x)}{2j+1} - \frac{\mathcal{T}_{2j-1}(x)}{2j-1}\right)\right).$$

By regrouping the terms, we get the following explicit form of the Chebyshev series of $\mathrm{erf}(\kappa x)$:

$$\mathrm{erf}(\kappa x) = \frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\sum_{n=0}^{\infty}(-1)^n\frac{I_n(\kappa^2/2)+I_{n+1}(\kappa^2/2)}{2n+1}\mathcal{T}_{2n+1}(x). \tag{15}$$

Now, to bound the coefficient norm, we use the following inequality from Reference [3]:

$$e^{-x}x^{-n}(I_n(x)+I_{n+1}(x)) \le \sqrt{\frac{2}{\pi}}\left(x+\frac{n}{2}+\frac{1}{4}\right)^{-n-\frac{1}{2}}.$$

Note that $I_n(x) \ge 0$ for $x \ge 0$ and all $n \in \mathbb{N}$. So, the above in fact bounds the absolute value of the left-hand side. We use this inequality to bound the (absolute value of the) coefficient of $\mathcal{T}_{2n+1}(x)$ in Equation (15) as follows:

$$\frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\frac{I_n(\kappa^2/2)+I_{n+1}(\kappa^2/2)}{2n+1} \le \frac{4}{\pi}\frac{1}{2n+1}\left(\frac{\kappa^2}{\kappa^2+n+1/2}\right)^{n+1/2}. \tag{16}$$

Using this inequality, we can bound the coefficient norm of the truncated Chebyshev series:

Lemma 24. *Let $N > 0$ be an integer. Then,*

$$\frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\sum_{n=0}^{N}\frac{I_n(\kappa^2/2)+I_{n+1}(\kappa^2/2)}{2n+1} \le \frac{6+2\log N}{\pi}.$$

Proof. Using Equation (16) and the fact that $0 \le \frac{\kappa^2}{\kappa^2+n+1/2} \le 1$, we get

$$\begin{aligned}\frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\sum_{n=0}^{N}\frac{I_n(\kappa^2/2)+I_{n+1}(\kappa^2/2)}{2n+1} &\le \frac{4}{\pi}\sum_{n=0}^{N}\frac{1}{2n+1}\left(\frac{\kappa^2}{\kappa^2+n+1/2}\right)^{n+1/2}\\ &\le \frac{4}{\pi}\sum_{n=0}^{N}\frac{1}{2n+1}.\end{aligned}$$

It is well-known that the last sum is $O(\log N)$. To be more precise,

$$\frac{4}{\pi}\sum_{n=0}^{N}\frac{1}{2n+1} \le \frac{4}{\pi}\left(1+\sum_{n=1}^{N}\frac{1}{2n}\right) = \frac{4}{\pi}+\frac{2}{\pi}\sum_{n=1}^{N}\frac{1}{n} \le \frac{4}{\pi}+\frac{2}{\pi}(1+\log N) \le \frac{6+2\log N}{\pi}.$$

□

Now, if we just want to bound the coefficients' 1-norm, then it suffices to take $N = \lceil\kappa^2\rceil$ and bound the rest of the coefficients using the following simple tail bound:

Lemma 25. *Let $N \ge \kappa^2$ be an integer. Then,*

$$\frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}}\sum_{n=N}^{\infty}\frac{I_n(\kappa^2/2)+I_{n+1}(\kappa^2/2)}{2n+1} \le 2^{2-N}.$$

Proof. Again, we start by using Equation (16), but now we note that for $n \geq \kappa^2$, $0 \leq \frac{\kappa^2}{\kappa^2+n+1/2} \leq \frac{1}{2}$, so

$$\frac{2\kappa e^{-\kappa^2/2}}{\sqrt{\pi}} \sum_{n=N}^{\infty} \frac{I_n(\kappa^2/2) + I_{n+1}(\kappa^2/2)}{2n+1} \leq \frac{4}{\pi} \sum_{n=N}^{\infty} \frac{2^{-n}}{2n+1} \leq \frac{2^{3-N}}{\pi} \leq 2^{2-N}. \qquad \square$$

Therefore, the coefficient norm of the entire series is bounded by $\frac{6+2\log\kappa^2}{\pi} + 2^{2-\kappa^2} \leq 4 + 2\log\kappa$. An easy consequence of this bound is that we can approximate $\operatorname{erf}(\kappa x)$ up to error $0 \leq \varepsilon \leq 2^{2-\kappa^2}$ with a polynomial of degree $\log_2(4/\varepsilon)$.

If the desired error $\varepsilon$ is larger than $2^{2-\kappa^2}$, then a more careful analysis of the tail bound for $\kappa \leq N \leq \kappa^2$ yields an $\varepsilon$-approximation polynomial of degree $O(\kappa\sqrt{\log(\kappa/\varepsilon)})$.

Lemma 26. *Let $1 \leq \alpha \leq \kappa$ be an integer. Then,*

$$\frac{4}{\pi} \sum_{n=\alpha\kappa}^{(\alpha+1)\kappa-1} \frac{1}{2n+1} \left(\frac{\kappa^2}{\kappa^2+n+1/2}\right)^{n+1/2} \leq \frac{4}{\pi} e^{\alpha^2/2}.$$

Proof. First, we note that $\left(\frac{\kappa^2}{\kappa^2+n+1/2}\right)^{n+1/2} \leq \left(\frac{\kappa^2}{\kappa^2+n}\right)^n$, so we get

$$\begin{aligned}
\frac{4}{\pi} \sum_{n=\alpha\kappa}^{(\alpha+1)\kappa-1} \frac{1}{2n+1} \left(\frac{\kappa^2}{\kappa^2+n+1/2}\right)^{n+1/2} &\leq \frac{4}{\pi} \sum_{n=\alpha\kappa}^{(\alpha+1)\kappa-1} \frac{1}{2n+1} \left(\frac{\kappa^2}{\kappa^2+n}\right)^n \\
\leq \frac{4}{\pi} \sum_{n=\alpha\kappa}^{(\alpha+1)\kappa-1} \frac{1}{2(\alpha\kappa)+1} \left(\frac{\kappa^2}{\kappa^2+\alpha\kappa}\right)^{\alpha\kappa} &= \frac{4}{\pi} \frac{\kappa}{2(\alpha\kappa)+1} \left(\frac{\kappa}{\kappa+\alpha}\right)^{\alpha\kappa} \\
= \frac{4}{\pi} \frac{\kappa}{2(\alpha\kappa)+1} \left(\frac{1}{1+\alpha/\kappa}\right)^{\alpha\kappa} &\leq \frac{4}{\pi} \frac{\kappa}{2(\alpha\kappa)+1} \left(e^{-\alpha/(2\kappa)}\right)^{\alpha\kappa} \\
= \frac{4}{\pi} \frac{\kappa}{2(\alpha\kappa)+1} e^{-\alpha^2/2} &\leq \frac{4}{\pi} e^{-\alpha^2/2}.
\end{aligned}$$

The second to last inequality requires $\alpha \leq \kappa$. $\square$

So, to get an $\varepsilon$-approximation polynomial, we just need to find an integer $1 \leq \alpha_0 \leq \kappa$ such that

$$2^{2-\kappa^2} + \frac{4}{\pi} \sum_{\alpha=\alpha_0}^{\kappa} e^{-\alpha^2/2} \leq \varepsilon.$$

Indeed, if we let $\varepsilon' = \varepsilon - 2^{2-\kappa^2}$, then it suffices to choose $\alpha_0 = \left\lceil \sqrt{2\log(\frac{4\kappa}{\pi\varepsilon'})} \right\rceil$, so we get

$$2^{2-\kappa^2} + \frac{4}{\pi} \sum_{\alpha=\alpha_0}^{\kappa} e^{-\alpha^2/2} \leq 2^{2-\kappa^2} + \frac{4\kappa}{\pi} e^{-\alpha_0^2/2} \leq 2^{2-\kappa^2} + \varepsilon' = \varepsilon.$$

Thus, the degree of the $\varepsilon$-approximating polynomial is $\alpha_0\kappa - 1 = O(\kappa\sqrt{\log(\kappa/\varepsilon)})$.

### A.3 Hamiltonian Simulation, $\sin(\kappa x)$ and $\cos(\kappa x)$

The Hamiltonian simulation problem requires one to evaluate the function $e^{i\kappa x}$ for a possibly large value of $\kappa$. This function is closely related to $\sin(\kappa x)$ and $\cos(\kappa x)$, and in this section, we focus on the case of $\cos(\kappa x)$ with $\kappa \geq 1$. We recall the Jacobi-Anger expansion [23, Equation (10.12.3)] of $\cos(\kappa x)$:

$$\cos(\kappa x) = J_0(\kappa) + 2\sum_{j=1}^{\infty}(-1)^j J_{2j}(\kappa)\mathcal{T}_{2j}(x), \tag{17}$$

where $J_n(x)$ is the Bessel function of the first kind. One could attempt to show an $\Omega(\sqrt{\kappa})$ lower bound on the coefficient 1-norm by using the large-argument approximation [23, Equation (10.7.8)]

$$J_n(x) \approx \sqrt{\frac{2}{\pi x}}\cos\left(x - \frac{n\pi}{2} - \frac{\pi}{4}\right).$$

Unfortunately this approximation is only valid for $n \ll \sqrt{x}$, or equivalently, $j \ll \sqrt{\kappa}$ (a simple proof would require the same equality up to $j = O(\kappa)$). Therefore, we turn to positive results and show how to evaluate $\cos(\kappa x)$ as a composition of easily-implementable functions corresponding to simple quantum circuits in the matrix (block-encoding) case.

First, we show that $\cos(x)$ can be approximated using a polynomial with a coefficient norm that is less than 1. We do this by showing that the 1-norm of the entire Chebyshev series of $\cos(x)$ is 1. By specializing the expansion Equation (17) for $\kappa = 1$ and observing that $\mathcal{T}_{2j}(0) = (-1)^j$, we get that

$$1 = \cos 0 = J_0(1) + 2\sum_{j=1}^{\infty} J_{2j}(1),$$

so to bound the coefficient norm, it suffices to show that $J_{2j}(1) \geq 0$ for all $j > 0$. This holds by Reference [23, Equation (10.14.2)] and Reference [23, Equation (10.14.7)], i.e.,

$$0 < J_\nu(\nu) < \frac{2^{\frac{1}{3}}}{3^{\frac{2}{3}}\Gamma\left(\frac{2}{3}\right)\nu^{\frac{1}{3}}} \quad \text{and} \quad 1 \leq \frac{J_\nu(\nu x)}{x^\nu J_\nu(\nu)} \leq e^{\nu(1-x)}.$$

To approximate $\cos(x)$ to an error $\varepsilon$, it suffices to truncate the series Equation (17) after $O(\log(1/\varepsilon))$ terms, since Reference [23, Equation (10.14.4)] gives us the tail bound $|J_n(x)| \leq \frac{|x|^n}{2^n n!}$.

Finally, for $\kappa > 1$ define $k := \lceil \kappa \rceil$, and note that it suffices to evaluate $\cos(kx)$ due to the (trivial) identity $\cos(\kappa x) = \cos(k(\kappa x/k))$. We conclude by observing that $\cos(kx) = \mathcal{T}_k(\cos x)$. The key point is that the functions $\kappa x/k$, $\cos(x)$, and $\mathcal{T}_k(x)$ can all be approximated with polynomials of coefficient norm at most 1, so composing them does not incur any costly subnormalization. In other words, by composing these functions, we can start from a 1-block-encoding of a matrix $A$ and construct 1-block-encodings of $\frac{\kappa}{k}A$, $\cos(\frac{\kappa}{k}A)$, and finally $\mathcal{T}_k(\cos(\frac{\kappa}{k}A)) = \cos(\kappa A)$.

We can apply a similar trick to evaluate $\sin(\kappa x)$. As before, we note that it suffices to implement $\sin(kx)$ for $k = \lceil \kappa \rceil$, and we use the following identity involving the $\mathcal{U}_k$, the Chebyshev polynomials of the second kind:

$$\sin(kx) = \mathcal{U}_{k-1}(\cos x)\sin(x).$$

An easy calculation shows that simply negating the ancilla qubit at the start of the circuit in Section 3.2 allows us to compute a 1-block-encoding of $\sin(\kappa A)$ given a 1-block-encoding of $\sin(A)$. Adding $\cos(\kappa A)$ and $\sin(\kappa A)$ using LCU yields a 2-block-encoding of $e^{i\kappa A}$.

What remains to be shown is that $\sin(A)$ can be evaluated efficiently using LCU, i.e., that its Chebyshev coefficient 1-norm is bounded by 1. We use the Jacobi-Anger expansion

$$\sin(x) = 2\sum_{j=0}^{\infty}(-1)^j J_{2j+1}(1)\mathcal{T}_{2j+1}(x),$$

so we need to bound the series $\sum_{j=0}^{\infty} |J_{2j+1}(1)|$ by 1/2. Using the bound $|J_n(x)| \leq \frac{|x|^n}{2^n n!}$, we can show that $\sum_{j=1}^{\infty} |J_{2j+1}(1)| \leq 1/36 < 0.03$, and we can calculate the numerical value $|J_1(1)| < 0.45$, which concludes the proof.

## REFERENCES


[1] Andris Ambainis. 2012. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *29th International Symposium on Theoretical Aspects of Computer Science (STACS'12) (Leibniz International Proceedings in Informatics (LIPIcs))*, Christoph Dürr and Thomas Wilke (Eds.), Vol. 14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 636–647. DOI: https://doi.org/10.4230/LIPIcs.STACS.2012.636

[2] Dong An and Lin Lin. 2022. Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm. *ACM Trans. Quant. Comput.* 3, 2 (2022), 1–28. DOI: https://doi.org/10.1145/3498331

[3] Árpád. Baricz and Tibor K. Pogány. 2014. On a sum of modified bessel functions. *Mediterr. J. Math.* 11, 2 (2014), 349–360. DOI: https://doi.org/10.1007/s00009-013-0365-y

[4] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. 2001. Quantum lower bounds by polynomials. *J. ACM* 48, 4 (2001), 778–797. DOI: https://doi.org/10.1145/502090.502097

[5] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. 2015. Hamiltonian simulation with nearly optimal dependence on all parameters. In *IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 792–809. DOI: https://doi.org/10.1109/FOCS.2015.54

[6] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. 2019. The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation. In *46th International Colloquium on Automata, Languages, and Programming (ICALP'19) (Leibniz International Proceedings in Informatics (LIPIcs))*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.), Vol. 132. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 33:1–33:14. DOI: https://doi.org/10.4230/LIPIcs.ICALP.2019.33

[7] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, and Mario Szegedy. 2020. Finding Angles for Quantum Signal Processing with Machine Precision. DOI: https://doi.org/10.48550/arXiv.2003.02831

[8] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. 2017. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* 46, 6 (2017), 1920–1950. DOI: https://doi.org/10.1137/16M1087072

[9] Pedro C. S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. 2021. Optimal Scaling Quantum Linear Systems Solver via Discrete Adiabatic Theorem. DOI: https://doi.org/10.48550/arXiv.2111.08152

[10] Yulong Dong, Xiang Meng, K. Birgitta Whaley, and Lin Lin. 2021. Efficient phase-factor evaluation in quantum signal processing. *Phys. Rev. A* 103, 4 (2021), 042419. DOI: https://doi.org/10.1103/PhysRevA.103.042419

[11] Sebastian Dörn and Thomas Thierauf. 2009. The quantum query complexity of the determinant. *Inform. Process. Lett.* 109, 6 (2009), 325–328. DOI: https://doi.org/10.1016/j.ipl.2008.11.006

[12] W. Morven Gentleman. 1972. Implementing Clenshaw-Curtis quadrature, II computing the cosine transformation. *Commun. ACM* 15, 5 (1972), 343–346. DOI: https://doi.org/10.1145/355602.361311

[13] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. 2019. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *51st Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 193–204. DOI: https://doi.org/10.1145/3313276.3316366

[14] Jeongwan Haah. 2019. Product decomposition of periodic functions in quantum signal processing. *Quantum* 3 (2019), 190. DOI: https://doi.org/10.22331/q-2019-10-07-190

[15] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* 103, 15 (2009), 150502. DOI: https://doi.org/10.1103/PhysRevLett.103.150502

[16] Iordanis Kerenidis and Anupam Prakash. 2020. Quantum gradient descent for linear systems and least squares. *Phys. Rev. A* 101, 2 (2020), 022316. DOI: https://doi.org/10.1103/PhysRevA.101.022316

[17] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. 2021. Quantum algorithms for second-order cone programming and support vector machines. *Quantum* 5 (2021), 427. DOI: https://doi.org/10.22331/q-2021-04-08-427

[18] Lin Lin and Yu Tong. 2020. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum* 4 (2020), 361. DOI: https://doi.org/10.22331/q-2020-11-11-361

[19] Noah Linden, Ashley Montanaro, and Changpeng Shao. 2020. Quantum vs. Classical Algorithms for Solving the Heat Equation. DOI: https://doi.org/10.48550/arXiv.2004.06516

[20] Guang Hao Low and Isaac L. Chuang. 2017. Hamiltonian Simulation by Uniform Spectral Amplification. DOI: https://doi.org/10.48550/arXiv.1707.05391

[21] Guang Hao Low and Isaac L. Chuang. 2019. Hamiltonian simulation by qubitization. *Quantum* 3 (2019), 163. DOI: https://doi.org/10.22331/q-2019-07-12-163

[22] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. 2021. Grand unification of quantum algorithms. *PRX Quantum* 2, 4 (2021), 040203. DOI: https://doi.org/10.1103/PRXQuantum.2.040203

[23] F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain (Eds.). 2022. *NIST Digital Library of Mathematical Functions* (release 1.1.5 ed.). Retrievedfrom:http://dlmf.nist.gov/

[24] Davide Orsucci and Vedran Dunjko. 2021. On solving classes of positive-definite quantum linear systems with quadratically improved runtime in the condition number. *Quantum* 5 (2021), 573. DOI: https://doi.org/10.22331/q-2021-11-08-573

[25] Ricardo Pachón and Lloyd N. Trefethen. 2009. Barycentric-Remez algorithms for best polynomial approximation in the chebfun system. *BIT Numer. Math* 49, 4 (2009), 721. DOI: https://doi.org/10.1007/s10543-009-0240-1

[26] Boris T. Polyak. 1987. *Introduction to Optimization.* Optimization Software, Publications Division.

[27] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. 2014. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* 113, 13 (2014), 130503. DOI: https://doi.org/10.1103/PhysRevLett.113.130503

[28] Theodore J. Rivlin. 1990. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory* (2nd ed.). Wiley.

[29] Sushant Sachdeva and Nisheeth K. Vishnoi. 2014. Faster algorithms via approximation theory. *Found. Trends® Theoret. Comput. Sci.* 9, 2 (2014), 125–210. DOI: https://doi.org/10.1561/0400000065

[30] Yiğit Subaşı, Rolando D. Somma, and Davide Orsucci. 2019. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.* 122, 6 (2019), 060504. DOI: https://doi.org/10.1103/PhysRevLett.122.060504

[31] Yu Tong, Dong An, Nathan Wiebe, and Lin Lin. 2021. Fast inversion, preconditioned quantum linear system solvers, fast green's-Function computation, and fast evaluation of matrix functions. *Phys. Rev. A* 104, 3 (2021), 032422. DOI: https://doi.org/10.1103/PhysRevA.104.032422

[32] Richard S. Varga. 2000. *Matrix Iterative Analysis.* Springer Series in Computational Mathematics, Vol. 27. Springer Berlin. DOI: https://doi.org/10.1007/978-3-642-05156-2

[33] Nisheeth K. Vishnoi. 2013. Lx = b. *Found. Trends Theoret. Comput. Sci.* 8, 1–2 (2013), 1–141. DOI: https://doi.org/10.1561/0400000054

Received 7 November 2022; revised 22 December 2023; accepted 13 February 2024