On solving classes of positive-definite quantum linear systems with quadratically improved runtime in the condition number

Orsucci, D.; Dunjko, V.

On solving classes of positive-definite quantum linear systems with quadratically improved runtime in the condition number

Davide Orsucci [1] and Vedran Dunjko [2]

[1] Institut für Kommunikation und Navigation, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Münchener Str. 20, 82234 Weßling, Germany

[2] Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

November 1, 2021

Quantum algorithms for solving the Quantum Linear System (QLS) problem are among the most investigated quantum algorithms of recent times, with potential applications including the solution of computationally intractable differential equations and speed-ups in machine learning. A fundamental parameter governing the efficiency of QLS solvers is κ, the condition number of the coefficient matrix A, as it has been known since the inception of the QLS problem that for worst-case instances the runtime scales at least linearly in κ [1]. However, for the case of positive-definite matrices classical algorithms can solve linear systems with a runtime scaling as $\sqrt{\kappa}$, a quadratic improvement compared to the the indefinite case. It is then natural to ask whether QLS solvers may hold an analogous improvement. In this work we answer the question in the negative, showing that solving a QLS entails a runtime linear in κ also when A is positive definite. We then identify broad classes of positive-definite QLS where this lower bound can be circumvented and present two new quantum algorithms featuring a quadratic speed-up in κ: the first is based on efficiently implementing a matrix-block-encoding of A^{-1}, the second constructs a decomposition of the form $A = LL^\dagger$ to precondition the system. These methods are widely applicable and both allow to efficiently solve BQP-complete problems.

1 Introduction

Quantum computation is described using the formalism of linear algebra, suggesting that quantum methods may be intrinsically well-suited to perform linear algebraic tasks. Algorithms solving linear systems of equations, in particular, are at the cornerstone of linear algebra [2, Chapter 2], having many direct applications and playing a pivotal role in several computational methods [3]. In the seminal work of Harrow, Hassadim, and Lloyd (HHL) the so-called Quantum Linear System (QLS) problem was introduced and a quantum algorithm was presented that allows solving the QLS exponentially faster than classical algorithms solving classical linear systems [1]. In subsequent works, several new algorithms have been put forward that solve QLS with further increased efficiency in comparison to the original HHL algorithm, improving the runtime dependence on the condition number [4], on the precision [5] and on the sparsity [6]. Recently, a new approach inspired by adiabatic quantum computation has introduced a significantly simpler quasi-optimal solving algorithm [7, 8], significantly narrowing the gap with experimental implementations [9], making the algorithm compatible with Near-term Intermediate Scale Quantum (NISQ) devices [10, 11] and leading to the development of the presently most efficient QLS solvers [12].

A key idea underpinning the possibility of achieving large quantum speed-ups in linear algebra tasks is the fact that an exponentially large complex vector can be compactly encoded in the amplitudes of a pure quantum state; e.g., a n-qubit state is described via 2^n amplitudes. This intuition is indeed correct for the QLS problem, which has been proven to be BQP-complete [1]:

any quantum computation can be re-formulated as a QLS with only a polynomial overhead and therefore there exist families of QLS problems that afford super-polynomial speed-ups compared to classical solution methods (unless BPP = BQP[1]). While this reduction shows that almost certainly there exist families of QLS problems that allow an exponential speed-up compared to all classical methods, the crucial question is whether there are *natural* problems that can be directly formulated and solved as QLS. The ubiquity of linear systems seems to suggests their quantum variant should be broadly applicable as well, but still it is not guaranteed, since in the QLS setting further significant constraints have to be met to obtain exponential speed-ups [13].

A prominent fundamental bottleneck of QLS solvers is that, to efficiently obtain the solution, it is not sufficient that the coefficient matrix A of the system $A\mathbf{x} = \mathbf{b}$ is invertible, but it also have to be *robustly* invertible, that is *well-conditioned*: it is required that the *condition number* of the matrix A, defined as the ratio between the largest and the smallest singular values, is small. In fact, solving a QLS necessarily entails a runtime scaling at least linearly in the condition number (unless BQP = PSPACE[2]) as was already proven in Ref. [1]. Therefore, an exponential quantum speed-up for QLS solving is achievable only when the condition number scales polylogarithmically with the system size and, unfortunately, it seems rather difficult to find natural examples of matrix families that exhibit such mild growth of the condition number [14]. However, polynomial quantum speed-ups for linear system solving should be rather broadly achievable and could still provide a quantum advantage if the degree of the polynomial is large enough [15]. In this view, obtaining a further quadratic improvement in the dependence on the condition number for some restricted classes of matrices, that are however of wide practical interest, could be of the uttermost importance for obtaining a broader impact of QLS solving algorithms. A previous publication showed that in the context of quantum algorithms for solving certain Markov chain problems the use of specialised QLS solvers for positive-definite matrices provides an improvement in the condition number dependence [16]. Exploring the general positive-definite QLS problem is the main focus of this work.

In the rest of the Introduction we motivate why quadratic speed-ups in the condition number in positive-definite QLS may be expected (Section 1.1), review some related results present in the literature (Section 1.2) and then proceed to give high-level overviews of our two algorithms (Section 1.3 and Section 1.4). In Section 2 we fix the notation and give the main definitions. In Section 3 we prove that QLS solvers require, even when restricting to positive-definite matrices, a runtime scaling linearly in the condition number. We then move to the main results of this work, that is, achieving improved runtime scaling for solving certain classes of positive-definite QLS: in Section 4 we show a method based on an efficient implementation of a matrix-block-encoding of A^{-1} and in Section 5 a method based on decomposing the coefficient matrix as $A = LL^\dagger$ to effectively precondition the system. Finally, an outlook of possible future research directions is given in Section 6.

1.1 Positive definite linear systems and quadratic speed-ups

In this work, we investigate the efficiency of QLS solving algorithms specialised to the case where the coefficient matrix $A \in \mathbb{C}^{N\times N}$ is Hermitian and positive definite (PD). We will call this sub-class the positive-definite Quantum Linear System (PD-QLS) problem.

A first reason to focus specifically on the PD case is that in the classical setting several problems of practical relevance are formulated as linear systems involving PD coefficient matrices. A second important reason is that the few fully worked-out examples in the literature providing strong evidence of polynomial quantum speed-ups for "natural" QLS problems involve PD coefficient matrices [14, 17]. In particular, the discretization of a partial differential equation (PDE) having a positive-definite kernel (such as, e.g., Poisson's equation) results in a large linear system where the coefficient matrix is positive definite. Montanaro and Pallister perform in Ref. [14] a detailed

[1] BPP is the class of decision problems that can be solved in bounded-error probabilistic polynomial time, BQP are those that can be solved with bounded-error polynomial time quantum computations. Loosely speaking, BPP = BQP would mean that the power of quantum computers is equal to that of classical computers.

[2] PSPACE is the class of decision problems that can be solved in classical polynomial space. With a Feymann sum-over-paths approach one can show that any quantum computation can be classically simulated in exponential time but with only polynomial space, thus BQP $\subseteq$ PSPACE. It is widely believed that BQP $\neq$ PSPACE.

Method	Result	Requirements
Reduction to majority problem	$\Omega(\kappa)$ query complexity lower bound Proposition 6	Access to A via a matrix-block-encoding or via a sparse-oracle access
Block-encoding of A^{-1}	$\mathcal{O}(\sqrt{\kappa})$ query and gate complexity Proposition 12 and Proposition 13	Access to normalised matrix-block-encoding of $B = (I - \eta A)$ and $\left\|\left\|A^{-1}\left\|\mathbf{b}\right\rangle\right\|\right\|$ is large
Decomposition as $A = LL^\dagger$	$\mathcal{O}(\sqrt{\kappa})$ gate complexity Proposition 16	A is the sum of PD local Hamiltonians, $\mathbf{b}$ is sparse and $1/\gamma$ in Eq. (91) is small

Table 1: Summary of the main results of this work. The results provided in the second column of the table hold under the conditions specified in the third column. The big-Ω notation is used for runtime lower bounds (in query complexity), the big-$\mathcal{O}$ notation for runtime upper bounds (exhibiting an explicit solving algorithm).

analysis of how QLS may be employed to solve the finite-element formulation of a PDE and show that the linear dependence on the condition number is the main bottleneck to obtaining large quantum speed-ups. In fact, the discretization of PDEs for functions defined in $\mathbb{R}^D$ typically results in positive-definite linear systems where the condition number scales as $\mathcal{O}\big(N^{2/D}\big)$ [14].

We highlight that one might have reasonably conjectured that it is possible to have a quadratically better scaling in κ, the condition number of A, in the PD case. First, note that the runtime lower bound of Ref. [1] is proven using a special family of matrices that are indefinite (neither positive nor negative definite) by construction, hence it is not directly applicable to the PD case. A standard method allows to transform an indefinite linear system into a PD one, but having a quadratically larger condition number[3]; hence, the lower bound in Ref. [1] directly yields a $\sqrt{\kappa}$ runtime lower bound for PD-QLS solvers. Second, the conjugate gradient (CG) descent method is the most efficient classical algorithm for solving PD linear systems and requires only $\mathcal{O}\big(\sqrt{\kappa}\log(1/\varepsilon)\big)$ iterations to converge to the correct solution, up to ε approximation. Each iteration consists in the update of some vectors having N entries, where N is the dimension of the linear system, thus resulting in a total runtime in $\mathcal{O}\big(N\sqrt{\kappa}\log(1/\varepsilon)\big)$ [18]. Then, it might seem plausible that a quantization of the CG method could yield a quantum algorithm having runtime in $\mathcal{O}\big(\mathrm{polylog}(N)\sqrt{\kappa}\log(1/\varepsilon)\big)$.

This conjectured quadratic speed-up is however not always achievable since, as we prove in Section 3, PD-QLS solvers have runtime scaling linearly in κ in worst-case problem instances. But this no-go result can be used as guidance to understand what is preventing us from achieving a better runtime scaling and, conversely, what additional conditions have to be imposed in order to achieve a quadratic speed-up in κ for PD-QLS solvers.

1.2 Previous related results

In context of general QLS solvers, i.e. solvers applicable also to indefinite or non-Hermitian matrices, the best algorithms have a runtime scaling quasi-linearly in κ, i.e. scaling as $\mathcal{O}\big(\kappa\,\mathrm{polylog}(\kappa)\big)$, thus almost saturating the linear lower bound [1]. Note that the original HHL algorithm has a worse performance, with a runtime scaling as $\mathcal{O}(\kappa^2/\varepsilon)$ where ε is the target precision. The first algorithm to achieve a quasi-linear scaling in κ was proposed by Ambainis in Ref. [4], which introduces a technique called Variable-Time Amplitude Amplification (VTAA) and employs it to optimize to the HHL algorithm. Subsequently, Childs, Kothari and Somma [5] introduced polynomial approximations of A^{-1} to exponentially improve the runtime dependence on the approximation error to $\mathcal{O}\big(\kappa^2\,\mathrm{polylog}(\kappa/\varepsilon)\big)$; they show, furthermore, that the VTAA-based optimization can be used also for this algorithm, thus yielding a $\mathcal{O}\big(\kappa\,\mathrm{polylog}(\kappa/\varepsilon)\big)$ runtime. Later, Chakraborty et al. showed that also the pseudo-inversion problem, whereby the matrix A may be non-invertible and even non-square, can be solved with a runtime in $\mathcal{O}\big(\kappa\sqrt{\gamma}\,\mathrm{polylog}(\kappa/\varepsilon)\big)$, where γ parametrises the overlap of $\mathbf{b}$ with the subspace where A is non-singular [19]. Finally, the current state-of-the-art methods for general QLS solving is given in Ref. [12], which does not rely on VTAA but instead is based

[3]Namely, for a given indefinite matrix A, the systems $A\mathbf{x} = \mathbf{b}$ and $A'\mathbf{x} = \mathbf{b}'$, with $\mathbf{b}' := A^\dagger\mathbf{b}$ and $A' := A^\dagger A$, have the same solution. The matrix A' is positive definite and $\kappa(A') = \kappa(A)^2$.

on ideas stemming from adiabatic quantum computation [7], which result in conceptually simpler algorithms and in a significant improvement of the polylogarithmic factors.

Furthermore, several specialized quantum algorithms have been introduced with the scope of more efficiently solving QLS for particular sub-classes of matrices. A few works, e.g. [17, 20, 21], have investigated the use of preconditioning to speed-up the solution. The main idea, which is well-known in classical linear system solving methods, is to look for an invertible matrix B, a so-called *preconditioner*, such that the matrix BA has a smaller condition number than A, and subsequently solve the equivalent linear system $BA = B\,\mathbf{b}$. An algorithm based on a sparse preconditioning matrix was introduced in Ref. [17], but it has very little formal guarantees of performance improvement. Another method based on circulant preconditioners was presented in Ref. [20], for which it is clearer how to assess when a runtime improvement can be achieved. Runtime improvements have been obtained in Ref. [21] applying new preconditioning methods to Hamiltonians arising in many-body physics. An entirely different approach, based on hybrid classical-quantum algorithms, has been explored in Ref. [22], which yields runtime speed-ups for the case where the rows or columns of the coefficient matrix can be prepared with polylogarithmic-depth quantum circuits. We also mention the result of Ref. [23], showing that it is possible to solve QLS for the special class of tridiagonal Toeplitz matrix with a runtime that scales polylogarithmically in all parameters, condition number included; note, however, that matrices of this class can be fully specified with just two real parameters. Finally, the PD-QLS problem has been previously considered in Ref. [16], where it is suggested that positive-definite systems could be solved more efficiently than indefinite ones.

1.3 Overview of the method based on a matrix-block-encoding of A^{-1}

Our first method for solving PD-QLS with improved runtime is based on implementing as a quantum circuit a unitary matrix $\mathcal{U}_{A^{-1}}$ that encodes in a sub-block a matrix proportional to A^{-1}, i.e. a so-called matrix-block-encoding [24, 25]. The solution state $\left|A^{-1}\mathbf{b}\right\rangle$ can be subsequently obtained via matrix-vector multiplication, achieved by applying the circuit encoding A^{-1} and projecting onto the correct sub-block. This method is analogous to the one introduced by Childs, Kothari and Somma in Ref. [5], where it is shown that exponentially precise polynomial approximations of the inverse function can be constructed, which then allow to implement a matrix-block-encoding of A^{-1} up to exponentially small error and finally solve the QLS problem via matrix-vector multiplication. Here, we show that if A is a PD matrix, an equally good approximation of A^{-1} can be obtained with polynomials having a quadratically smaller degree, leading to the possibility of a quadratic speed-up in PD-QLS solving.

More in detail, Ref. [5] considers a Hermitian matrix A whose spectrum is by assumption contained in the domain $\mathcal{D}_\Delta := [-1, -\Delta] \cup [+\Delta, +1]$, with $\Delta \leq 1/\kappa$. Then, families of real polynomials $P(x)$ are constructed such that $|P(x) - 1/x| \leq \varepsilon$ on the domain $\mathcal{D}_\Delta$ and have a degree $\ell \in \mathcal{O}\big(\kappa \log(1/\varepsilon)\big)$, see panel (a) of Figure 1 for an illustrative example. Using either the Linear Combination of Unitaries (LCU) [26, 27] or the Quantum Signal Processing (QSP) [28, 24, 25] method, it is possible to implement a matrix-block-encoding of $P(A)$, up to some rescaling factor $K > 0$ such that $P(A)/K$ can be encoded a sub-block of a unitary matrix, and then we have $\big|\big|P(A) - A^{-1}\big|\big| \leq \varepsilon$ in operator norm. The query complexities of LCU and QSP scale at least linearly in the degree of the polynomial, hence the use of polynomials of low degree is crucial to construct efficient QLS solvers. Moreover, the normalisation factor K of the matrix-block-encoding of $P(A)$ scales linearly in κ and enters multiplicatively in the cost of the matrix-vector multiplication step necessary to produce $\left|A^{-1}\mathbf{b}\right\rangle$.

In case A is a PD matrix, we can exploit the knowledge that its spectrum is contained in $\mathcal{D}^+_\Delta := [\Delta, 1]$ to perform the following trick: we assume to have access to a *normalised* matrix-block-encoding of $B := I - \eta\, A$, for some constant[4] $\eta \in (0, 2]$, so that the spectrum of B is always contained in the interval $\mathcal{D}_B = [-1, 1 - \eta\Delta]$. We then construct a polynomial $P(x)$ that approximates the function $f(x) := 1/(1-x)$ on the domain $\mathcal{D}_B$ up to ε distance. Using the QSP

[4] We show in Section 4.3 that B can be efficiently constructed for $\eta = 1$ when A is diagonally dominant and for $\eta = 1/J$ when A is the sum of J positive semi-definite local Hamiltonian terms.

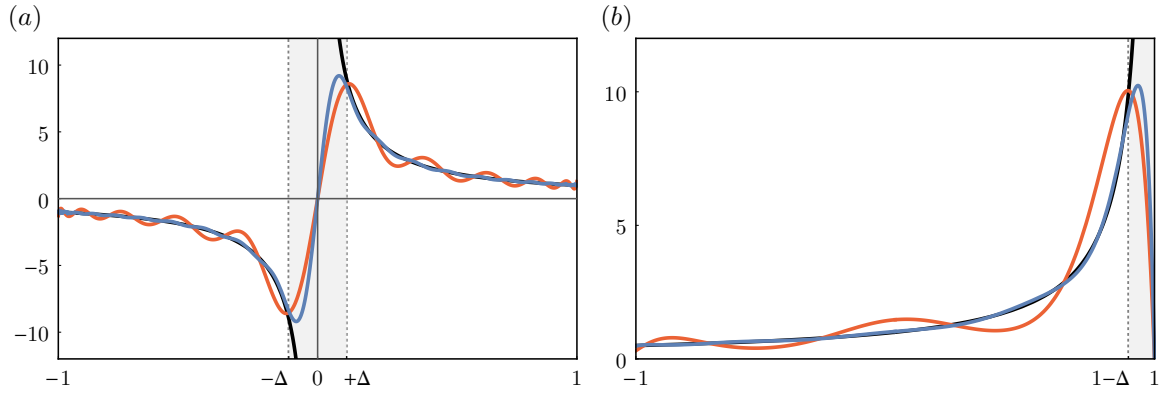

Figure 1: In panel (a) are shown two polynomials approximating the function $1/x$ (in black) chosen from the family of polynomials given in Ref. [5, Lemma 14], using as parameters $b = 200, j_0 = 10$ (red curve) and $b = 200, j_0 = 20$ (blue curve curve). In panel (b) are shown two polynomials approximating the function $1/(1-x)$ (in black) chosen from the family given in Eq. (26), using as parameters $\ell = 6, \kappa = 15$ (red curve) and $\ell = 10, \kappa = 9$ (blue curve). The shaded regions in each panel indicate the intervals where the polynomial approximation is not accurate.

method to implement a matrix-block-encoding of the matrix $P(B)$ we then have

$$P(B) \approx f(B) = \frac{1}{I - B} = \frac{1}{I - (I - \eta\, A)} = \frac{1}{\eta} A^{-1} \tag{1}$$

which is the required matrix inverse, up to a $1/\eta$ rescaling factor. Importantly, our polynomial approximation of $1/(1-x)$ has a degree $\ell \in \mathcal{O}\big(\sqrt{\kappa/\eta}\, \log(\kappa/(\eta\varepsilon))\big)$, a quadratically better dependence on κ with respect to the approximation of $1/x$ given in Ref. [5], see panel (b) of Figure 1. Moreover, the normalisation factor K scales again linearly in κ, which is the best dependence achievable.

From a mathematical standpoint, the possibility of a quadratic reduction of the degree of the approximating polynomial can be interpreted as a consequence of Bernstein's inequality [29]. This inequality states that in the class of real polynomials $P(x)$ of degree ℓ such that $|P(x)| \leq 1$ for all $x \in [-1, +1]$ the derivative $P'(x)$ satisfies $|P'(x)| \leq (1 - x^2)^{-1/2}\ell$ for all $-1 < x < 1$, while we have $|P'(x)| \leq \ell^2$ for $x = 1$ and $x = -1$ (and these last bounds are saturated by Chebyschev polynomials). Note that polynomials approximations of $1/x$ and of $1/(x-1)$ have high derivative close to the singularities, respectively in $x = 0$ and in $x = 1$, and because of Bernstein's inequality the latter case allows good polynomial approximations having a quadratically lower degree.

We need next to perform matrix-vector multiplication to obtain the state $\big|A^{-1}\mathbf{b}\big\rangle$ and thus solve the QLS; this is obtained by applying the quantum circuit that encodes the matrix $P(B) \propto A^{-1}$ to a quantum state of the form $|0\rangle\,|\mathbf{b}\rangle$ and then post-selecting the ancilla register to be in $|0\rangle$ or, more efficiently, using amplitude amplification [30]. The amplitude amplification step implies a multiplicative overhead of order $1/\kappa$ in the worst case, yielding a total runtime in $\mathcal{O}\big(\kappa^{3/2} \log(\kappa/\varepsilon)\big)$ for this PD-QLS solver. However, the efficiency the matrix-vector multiplication depends on the post-selection success probability and thus on the specific choice for the vector $\mathbf{b}$. In a best-case scenario the post-selection success probability is constant and thus the overall runtime of the PD-QLS solver is $\mathcal{O}\big(\sqrt{\kappa} \log(\kappa/\varepsilon)\big)$, while in the same high-success-probability scenario the QLS solver of Ref. [5] has a runtime in $\mathcal{O}\big(\kappa \log(\kappa/\varepsilon)\big)$, since this is the cost of implementing a polynomial approximation of A^{-1} for indefinite matrices.

We remark that there is a technical assumption that has to be satisfied to allow the realisation of our method, namely, that it is possible to implement a *normalised* matrix block encoding of $B = I - \eta\, A$. This is a non-trivial task: standard methods could allow us to prepare, for instance, a normalised matrix-block-encoding of B/d where $d \geq 2$ is the *sparsity* of B, i.e. the maximum number of non-zero entries in each column of B [5]. Note, however, that we would then need to implement an approximation of the function $g(x) := 1/(1 - xd)$ to obtain $g(B/d) = \frac{1}{\eta} A^{-1}$; the function $g(x)$ has a singularity in $x = 1/d$, which is in the interior of the domain $[-1, +1]$, and thus Bernstein's inequality precludes us from achieving good approximations with low-degree

polynomials for this function.

We will show that it is possible to implement normalised matrix-block encodings of B for two special classes of QLS problems: the first is when A is a diagonally dominant matrix; the second, when A is given as the sum of local PD Hamiltonian terms, where by "local" we mean that it acts non-trivially only on a small number of qubits. In these two cases it is therefore possible to implement our improved PD-QLS solver. We leave the question whether it is possible in broader generality to prepare normalised block-encodings as an open research question.

1.4 Overview of the method based on the $A = LL^\dagger$ decomposition

Our second method for solving PD-QLS with improved runtime is based on finding a decomposition $A = LL^\dagger$, akin to the classical Cholesky decomposition [31] which exists for all PD matrices, and then use L^{-1} as an efficient and effective preconditioner. Note, in fact, that $L^\dagger \mathbf{x} = \mathbf{b}'$ for $\mathbf{b}' = L^{-1}\mathbf{b}$ is a linear system equivalent to the original one, but the decomposition $A = LL^\dagger$ immediately implies $\kappa(L) = \kappa(L^\dagger) = \sqrt{\kappa(A)}$ and thus the new system provably has a quadratically smaller condition number. This decomposition is similar to a *spectral gap amplification* of A [32].

The method involves thus two main steps: in the first one we use classical computation to efficiently obtain a description of a matrix L such that $LL^\dagger = A$ and such that it is possible to efficiently find, using only classical computation, a description of the vector $\left|\mathbf{b}'\right\rangle := \left|L^{-1}\mathbf{b}\right\rangle$; in the second one, we use an efficient quantum algorithm, having runtime quasi-linear in κ, to solve $\left|L^\dagger \mathbf{x}\right\rangle = \left|\mathbf{b}'\right\rangle$, which thus gives $|\mathbf{x}\rangle = \left|(L^\dagger)^{-1}L^{-1}\mathbf{b}\right\rangle = \left|A^{-1}\mathbf{b}\right\rangle$. Note that the classical descriptions of $L^\dagger$ and of $\left|\mathbf{b}'\right\rangle$ should also allow the efficient compilation the quantum algorithm used to solve the preconditioned QLS.

The picture is not yet complete, since we actually use a matrix L that is non-square and thus singular. As a result, the inversion operation has to be substituted by a pseudo-inversion and the condition number by the *effective* condition number, the ratio between the largest and the smallest *non-zero* singular value; when A is invertible the effective condition number of L is quadratically smaller than the condition number of A. We also use two different pseudo-inverses in the classical and in the quantum part of the computation: in the quantum step the Moore-Penrose pseudo-inverse $(L^\dagger)^+$ is employed and in the classical preconditioning a *generalised* pseudo-inverse L^g chosen such that $(L^\dagger)^+ L^g = A^{-1}$; thus, they yield the desired solution $|\mathbf{x}\rangle = \left|A^{-1}\mathbf{b}\right\rangle$ when applied in sequence. These extensions to non-square matrices and to different pseudo-inverses are made to give leeway in the design of the classical part of the algorithm, allowing us to meet the efficiency requirements mentioned above. Finally, we will employ the QLS solver of [19], which can tackle pseudo-inversion problems and has a runtime quasi-linear in the effective condition number.

We show that a fully suitable decomposition of the form $A = LL^\dagger$ can be constructed for the Sum-QLS problem, i.e., when A is a sum of local PD Hamiltonian terms. In this case, the matrix L is formed by several blocks, each constructed from a single Hamiltonian term, while the pseudo-inverse L^g is obtained inverting the individual blocks, operations involving only small matrices and thus classically feasible. We also require that the vector $\mathbf{b}$ is sparse, containing only polynomially many non-zero entries, thus allowing to efficiently compute the description of $\left|\mathbf{b}'\right\rangle = |L^g\,\mathbf{b}\rangle$.

As a final technical caveat, we note that because of the mismatch between the pseudo-inverse used in the classical and in the quantum part of the algorithm (L^g and $(L^\dagger)^+$), the vector $\mathbf{b}'$ is not entirely contained in the support of $(L^\dagger)^+$ and thus amplitude amplification of the component in the correct subspace is required. This incurs in a multiplicative overhead of a factor $1/\sqrt{\gamma}$, where $\sqrt{\gamma} > 0$ is a known bound for the amplitude of the "good" component of $\mathbf{b}'$. This method thus has a provable runtime improvement over competing QLS solvers whenever $1/\sqrt{\gamma}$ is sufficiently small.

2 Notation and definitions

We assume knowledge of the main quantum computation concepts, as given for instance in Ref. [33]. A quantum computation is described using a Hilbert space of dimension 2^n for some n, corresponds to a system of n qubits, having a distinguished computational basis.

2.1 Linear algebra and asymptotic notation

We consistently denote with $N \in \mathbb{N}$ the dimension of the QLS we aim to solve and we define $n := \lceil \log_2 N \rceil$, so that a vector in $\mathbb{C}^N$ (possibly padded with zeroes at the end if N is not a power of two) can be described as pure state of n qubits. For any complex matrix $A \in \mathbb{C}^{N\times M}$ having N rows and M columns we write its Singular Value Decomposition (SVD) as $A = V\Sigma W^\dagger$ where V and W are unitary matrices of size $N \times N$ and $M \times M$ respectively and Σ is a real non-negative matrix of size $N \times M$ which is uniquely determined up to reordering of the diagonal entries and contains the singular values of A on the main diagonal. An Hermitian matrix A is positive definite if $\langle \mathbf{v}| A |\mathbf{v}\rangle > 0$ for all $|\mathbf{v}\rangle$ and is is positive semi-definite if $\langle \mathbf{v}| A |\mathbf{v}\rangle \geq 0$. The eigenvalues of A are real, positive (in the definite case) or non-negative (in the semi-definite case) and coincide with its singular values. For a general A, $\varsigma_{\min}, \varsigma_{\max}$ and $\lambda_{\min}, \lambda_{\max}$ denote the minimum and maximum singular values and eigenvalues, respectively. The Moore-Penrose pseudo-inverse of A is obtained by applying to the singular values ς_i of A the function $f : \mathbb{R} \mapsto \mathbb{R}$ defined as $f(x) = 1/x$ for $x \neq 0$ and $f(0) = 0$. More precisely, for a diagonal matrix Σ we define $\Sigma^+ = f(\Sigma)$, while for a general matrix $A = V\Sigma W^\dagger$ the pseudo-inverse is given as $A^+ := W\Sigma^+ V^\dagger$. Given $A \in \mathbb{C}^{N\times M}$ a *generalised* pseudo-inverse $A^g \in \mathbb{C}^{M\times N}$ is any matrix satisfying the equation $A\,A^g A = A$.

In this work we employ the ℓ^2-norm for vectors $||\mathbf{v}||^2 := \sum_{i=1}^N \mathrm{v}_i^2$ and the induced operator norm for matrices $||A|| := \max_{\mathbf{v}\neq 0} ||A\mathbf{v}|| \,/\, ||\mathbf{v}||$. The condition number of a matrix is given by $\kappa(A) := ||A||\,||A^{-1}||$. Since we have $||A|| = \varsigma_{\max}(A)$ and $||A^{-1}|| = 1/\varsigma_{\min}(A)$, the condition number can be also written as $\kappa(A) = \frac{\varsigma_{\max}(A)}{\varsigma_{\min}(A)}$. For a singular matrix A we define the *effective* condition number to be $\kappa_{\text{eff}}(A) := ||A||\,||A^+||$, which is equal to the ratio between the largest and the smallest non-zero singular value of A. A Hermitian matrix A is diagonally dominant if $\sum_{j:j\neq i} |A_{ij}| \leq |A_{ii}|$ for all i and note that $|A_{ii}| = A_{ii} > 0$ when A is positive definite.

We use the standard big-$\mathcal{O}$ and small-$\mathfrak{o}$ notations for asymptotic scaling, together with the following definitions: $f(x) \in \Omega(g(x))$ if and only if $g(x) \in \mathcal{O}(f(x))$, which is used to give lower bounds, and $\Theta(g(x)) = \mathcal{O}(g(x)) \cap \Omega(g(x))$. We also use the soft-$\mathcal{O}$ and soft-Ω notations where $f(x) \in \widetilde{O}(g(x))$ means $f(x) \in \mathcal{O}\big(g(x)\,\mathrm{polylog}\,[g(x)]\big)$, and similarly for $\widetilde{\Omega}(g(x))$, which are used to give more coarse-grained bounds.

2.2 Definition of the Quantum Linear System problem

In this section we introduce the main definitions that are relevant in the contest of the QLS problem, which is a quantum analogue of the classical linear algebra problem of solving the system of equations $A\mathbf{x} = \mathbf{b}$, having solution $\mathbf{x} = A^{-1}\mathbf{b}$ when A is invertible.

We use pure quantum states to encode N-dimensional complex vectors. A vector $\mathbf{v}$ enclosed in a bra or in a ket is always assumed to be normalised, $\big|\big|\,|\mathbf{v}\rangle\,\big|\big| = 1$. In particular we have:

$$|\mathbf{b}\rangle = \frac{\mathbf{b}}{||\mathbf{b}||} = \frac{\sum_{i=1}^N b_i\,|i\rangle}{\left(\sum_{i=1}^N |b_i|^2\right)^{1/2}} \tag{2}$$

$$\left|A^{-1}\mathbf{b}\right\rangle = \frac{A^{-1}\,|\mathbf{b}\rangle}{||A^{-1}\,|\mathbf{b}\rangle||}\,. \tag{3}$$

We now give a general definition of the QLS problem. The formulation is similar to the one provided in Ref. [7] and is intentionally not specifying the access models employed for the coefficient matrix A and the known-term vector $\mathbf{b}$, for sake of generality.

Definition 1 (Quantum Linear System)**.** *Suppose we have access to a vector $\boldsymbol{b} \in \mathbb{C}^N \setminus \{0\}$ and to a non-singular matrix $A \in \mathbb{C}^{N\times N}$ (access is given via quantum oracles, or some kind of implicit or explicit description). We are given two real positive parameters ς_* and ς^* such that $\varsigma_* \leq \varsigma_{\min}(A)$ and $\varsigma_{\max}(A) \leq \varsigma^*$, i.e. the singular values of A are contained in the interval $\mathcal{D}_A = \big[\varsigma_*, \varsigma^*\big]$; equivalently, we are given two parameters $\kappa > 1$ and $\alpha > 0$ that provide the upper bounds $\kappa(A) \leq \kappa$ and $||A|| \leq \alpha$. We are also given a target precision $\varepsilon > 0$.*

The QLS problem then consists in preparing a density matrix $\rho_{\boldsymbol{x}}$ which is ε-close in trace distance to the solution vector $|\boldsymbol{x}\rangle = \left|A^{-1}\boldsymbol{b}\right\rangle$; that is, we require that

$$\big|\big|\,\rho_{\boldsymbol{x}} - |\boldsymbol{x}\rangle\langle\boldsymbol{x}|\,\big|\big|_{\mathrm{Tr}} \leq \varepsilon \tag{4}$$

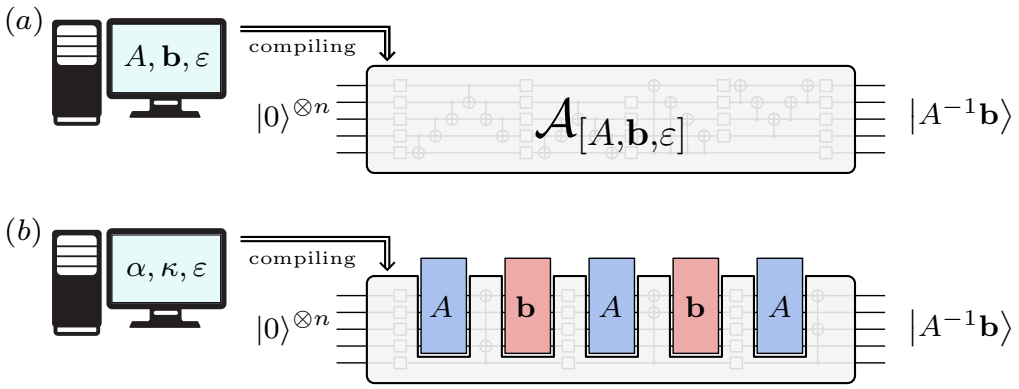

Figure 2: Different access models for QLS algorithms. Panel (a) illustrates the case where a classical description of A and $\mathbf{b}$ (and the target precision ε) is provided and then used to compile a quantum algorithm $\mathcal{A}$, solving the QLS for the given A and $\mathbf{b}$. The description of A and $\mathbf{b}$ does not need to be fully explicit: it is sufficient that it allows to efficiently compute the sequence of elementary quantum gates of $\mathcal{A}$. Panel (b) illustrates the case of a *relativising* quantum algorithm; in this case, the sequence of elementary quantum gates only depends on a few parameters (e.g., $\alpha, \kappa, \varepsilon$ as in Definition 1) while two fixed sub-routines specify A and $\mathbf{b}$ and these sub-routines can be treated as black-boxes.

where the trace norm is defined as $||X||_{\mathrm{Tr}} := \frac{1}{2}\mathrm{Tr}\left(\sqrt{XX^\dagger}\right)$.

Definition 2 (Positive-definite Quantum Linear System)**.** *A PD-QLS problem is a QLS problem, as given in Definition 1, where the coefficient matrix A is Hermitian and positive definite.*

We note that a more commonly employed definition requires that the QLS solver outputs a state $|\widetilde{\mathbf{x}}\rangle$ such that $\left|\left|\, |\widetilde{\mathbf{x}}\rangle - |\mathbf{x}\rangle \,\right|\right| \leq \varepsilon$. We prefer to use the trace distance since it is operationally motivated, as it equals the probability of distinguishing two copies of two quantum states when optimizing over all possible measurements, see [33, Section 9.2.1]. The trace distance then also bounds the maximum relative error that can be introduced when estimating the expectation value $\mathrm{Tr}(|\mathbf{x}\rangle\langle\mathbf{x}|\, M)$ for a given observable M and computing expectation values was the end goal of Ref. [1]. Finally, for pure normalised states $|\psi\rangle$ and $|\phi\rangle$ the trace distance simplifies as $d_{\mathrm{Tr}}(\psi,\phi) := \left|\left|\, |\psi\rangle\langle\psi| - |\phi\rangle\langle\phi| \,\right|\right|_{\mathrm{Tr}} = \sqrt{1-|\langle\psi|\phi\rangle|^2}$ and using $|\langle\psi|\phi\rangle| \geq 1 - \frac{1}{2}\left|\left|\, |\psi\rangle - |\phi\rangle \,\right|\right|^2$ we obtain the inequality

$$d_{\mathrm{Tr}}(\psi,\phi) \leq \left|\left|\, |\psi\rangle - |\phi\rangle \,\right|\right|, \tag{5}$$

thus the trace distance between $|\widetilde{\mathbf{x}}\rangle$ and $|\mathbf{x}\rangle$ is at least as small as their ℓ^2 distance.

It is customary to assume that A has been rescaled by a factor $\alpha \geq ||A||$, so that we take, without loss of generality, $||A|| \leq 1$ and the only parameter that needs to be specified is an upper bound to the condition number. This will be also our convention, unless otherwise specified.

2.3 Oracles for quantum linear system solving

We now define and discuss a few different access models for A and $\mathbf{b}$, since the results we present for the PD-QLS solvers crucially depend on which access model is assumed. A fundamental distinction is between oracular and non-oracular (a.k.a. relativising and non-relativising) algorithms, see Figure 2. For instance, in oracular settings it is often possible to establish unconditional query complexity lower bounds, while in non-oracular settings non-trivial runtime lower bounds typically can be proven only under some (reasonable) complexity theory assumptions, such as $\mathsf{P} \neq \mathsf{NP}$. Note that most of the literature on QLS assumes oracular access to A and $\mathbf{b}$ [1, 4, 5, 6, 7, 12, 19].

We start defining the access model for $\mathbf{b}$ that we assume throughout this paper, except where differently specified.

Definition 3 (State preparation oracle, as in [1])**.** *Given a vector $\boldsymbol{b} \in \mathbb{C}^N$ we say that we have quantum access to a* state preparation oracle *for $\boldsymbol{b}$ if there is a unitary operator $\mathcal{U}_{\boldsymbol{b}}$ such that $\mathcal{U}_{\boldsymbol{b}}|0^n\rangle = |\boldsymbol{b}'\rangle$, where $\boldsymbol{b}'$ is obtained by padding $\boldsymbol{b}$ with zeroes until its size is a power of 2.*

As already noted by HHL, a general setting where it is possible to efficiently implement $\mathcal{U}_B$ is the one presented by Grover and Rudolph [34]; another possibility is that $\mathcal{U}_\mathbf{b}$ is directly encoded in a qRAM [35], as may be required in quantum machine learning contexts.

Next, we define two models for access to A, which we denote as $\mathcal{P}_A$ and $\mathcal{U}_A$ and which correspond to a sparse-matrix-access and matrix-block-encoding, respectively.

Definition 4 (Sparse-matrix-access, as in [5])**.** *Given a Hermitian matrix*[5] *A which is d-sparse (i.e., has at most d non-zero entries in each row and column) a quantum* sparse-matrix-access *$\mathcal{P}_A$ is given by a pair of oracular functions*

$$\mathcal{P}_A := (\mathcal{P}_A^{pos}, \mathcal{P}_A^{val}) \tag{6}$$

where $\mathcal{P}_A^{pos}$ and $\mathcal{P}_A^{val}$ specify the positions of the (potentially) non-zero entries of A and the values of those entries, respectively, i.e.

$$\mathcal{P}_A^{pos} : |i,\nu\rangle \;\mapsto\; |i, j(i,\nu)\rangle \qquad \forall\; i,j \in \{1,\dots,N\} \text{ and } \nu \in \{1,\dots,d\} \tag{7}$$

$$\mathcal{P}_A^{val} : |i,j,z\rangle \mapsto |i,j,A_{i,j}\oplus z\rangle \qquad \forall\; z \in \{0,1\}^* \tag{8}$$

where $A_{i,j}\oplus z \in \{0,1\}^*$ denotes a bit string of arbitrary length that encodes the value $A_{i,j} \in \mathbb{C}$.

In order to keep the presentation simple, we assume here and throughout the paper that numeric representations of complex numbers can be specified exactly or with a sufficiently high number of digits of precision.

Definition 5 (Matrix block encoding, as in [28, 25])**.** *A unitary operator $\mathcal{U}_A$ acting on $n+a$ qubits is called an (α, a, ε)-matrix-block-encoding of a n-qubit operator A if* [6]

$$\left|\left| A - \alpha \left(\langle 0^a| \otimes I\right) \mathcal{U}_A \left(|0^a\rangle \otimes I\right) \right|\right| \leq \varepsilon \tag{9}$$

which can be expressed also as:

$$\mathcal{U}_A = \begin{pmatrix} \widetilde{A}/\alpha & * \\ * & * \end{pmatrix} \quad \text{with} \quad \left|\left|\widetilde{A} - A\right|\right| \leq \varepsilon\,, \tag{10}$$

where the asterisks ($*$) denote arbitrary matrix blocks of appropriate dimensions.

We call α the normalization factor *of the matrix-block-encoding and we say in the special case where $\alpha = 1$ that $\mathcal{U}_A$ is a* normalised *matrix-block-encoding of A.*

A technique introduced by Childs allows to implement a $(d, 1, 0)$-matrix-block-encoding of A, where the normalisation constant d is equal to the sparsity of A, using only a constant number of accesses to $\mathcal{P}_A$ and $\mathcal{O}\big(\text{poly}(n)\big)$ extra elementary gates, see Ref. [5] and references therein. In short, we have the reduction:

$$\mathcal{P}_A \implies \mathcal{U}_A \tag{11}$$

where the arrows means that having access to an oracle of first type allows to efficiently implement the oracle of the second type.

We also note that other access models to A have been considered in the literature; for example in Ref. [36] it is assumed that is possible to efficiently prepare quantum states that are proportional to each one of the columns of A.

[5] Since A is Hermitian, access by rows and by columns are equivalent. The definition can be extended to non-Hermitian matrices, but we need to assume access both by rows and by columns.

[6] The circuit $\mathcal{U}_A$ may also act on other ancilla qubits, which are in $|0\rangle$ both before and after the application of $\mathcal{U}_A$. For a given $(\cdot, a, \cdot)$-matrix-block-encoding we only count the a ancilla qubits that require post-selection to $\langle 0^a|$ to obtain the encoding of A.

2.4 Quantum linear systems in non-oracular settings

We consider in Section 5 (and also briefly in Section 4.3) a case that we call the Sum-of-Hamiltonians QLS (Sum-QLS) problem, which is not formulated as an oracular algorithm but is based, instead, on a classical description of A and $\mathbf{b}$. In order to obtain efficient Sum-QLS solving algorithms, it is thus necessary that the descriptions of A and $\mathbf{b}$ are compact, depending at most on $\mathcal{O}\big(\mathrm{poly}(n)\big)$ real or complex parameters.

For the known term vector $\mathbf{b}$, we will simply assume that it is a sparse vector in the computational basis, with at most $\mathcal{O}\big(\mathrm{poly}(n)\big)$ non-zero entries, whose position is also known. Hence, a fully explicit classical description of $\mathbf{b}$ can be provided and this also enables efficiently implementing a state preparation circuit $\mathcal{U}_{\mathbf{b}}$.

For the coefficient matrix A we give a more implicit description: the entries of A are not specified one-by-one (which would be inefficient, as the matrix size $N \in \Theta(2^n)$ is assumed to be very large) but rather can be computed from only a relatively small set of parameters, scaling polynomially in the number of qubits. Specifically, we assume that A is given as the sum of polynomially many local Hamiltonian terms:

$$A = \sum_{j=1}^{J} H_{(j)} \qquad \forall j \ H_{(j)} \text{ is positive (semi)-definite,} \tag{12}$$

where the number of terms is $J \in \mathcal{O}\big(\mathrm{poly}(n)\big)$ and each Hamiltonian $H_{(j)}$ acts on a small number of qubits, namely, on at most $\mathcal{O}\big(\log(n)\big)$ qubits. This case has been previously considered in Ref. [16].

3 Query complexity lower bound

In this section we prove a $\Omega(\kappa)$ lower bound on the runtime of QLS which is alternative to the ones given by HHL in Ref. [1]. The main innovation we introduce is that our lower bound applies also to the PD-QLS case, while the proofs given by HHL only yield a $\widetilde{\Omega}(\sqrt{\kappa})$ lower bound when specialised to PD matrices. More precisely, we have the following result.

Proposition 6 (Query complexity lower bound)**.** *Consider oracular quantum algorithms that solve the PD-QLS problem as presented in Definition 2 for different access models to A and $\mathbf{b}$. Namely, access to $\mathbf{b}$ is given via a state preparation oracle $\mathcal{U}_{\mathbf{b}}$ (Definition 3), while access to A is given either via a sparse-matrix oracle $\mathcal{P}_A$ (Definition 4) or via a matrix-block-encoding $\mathcal{U}_A$ (Definition 5). Then, PD-QLS solving algorithms reaching a constant precision $\varepsilon \in \mathcal{O}(1)$ have query complexities $Q[\mathcal{U}_{\mathbf{b}}], Q[\mathcal{U}_A], Q[\mathcal{P}_A]$ all in $\Omega\big(\min(\kappa, N)\big)$.*

The proof of these lower bounds is rather technical and can be found in Appendix A. We now present a weaker result that, however, can be easily proven as a consequence of the optimality of Grover search [37]; namely, we show that PD-QLS solving has a linear scaling in κ for all $\kappa \leq \sqrt{N}$.

Proposition 7. *Consider a PD-QLS problem as presented in Definition 2 and suppose that access to A is given by a sparse-matrix oracle $\mathcal{P}_A$ (Definition 4), with no assumption on the access model for $\mathbf{b}$. Then, a quantum algorithm that solves the QLS up to any constant precision $\varepsilon \in [0, 1)$ must make $\Omega\big(\min(\kappa, \sqrt{N})\big)$ accesses to $\mathcal{P}_A$.*

Proof. Consider the search problem of finding an element $j \in \mathcal{S}$, where $\mathcal{S} \subseteq \{1, \ldots, N\}$ is a subset containing M elements. The membership of a element j in $\mathcal{S}$ is encoded as a quantum oracle $\mathcal{P}_{\mathcal{S}}$ which flips a ancilla qubit if $j \in \mathcal{S}$ and leaves the ancilla unchanged if $j \notin \mathcal{S}$.

Consider, next, a matrix A that is diagonal (and thus 1-sparse) having entries

$$A_{j,j} \ = \ \begin{cases} \alpha = \sqrt{\frac{N-M}{N}} & \text{if } j \notin \mathcal{S} \\ \beta = \sqrt{\frac{M}{N}} & \text{if } j \in \mathcal{S}\,. \end{cases} \tag{13}$$

The sparse-matrix oracle $\mathcal{P}_A = (\mathcal{P}_A^{pos}, \mathcal{P}_A^{val})$ for this diagonal matrix A can be implemented with exactly one access to the membership oracle $\mathcal{P}_{\mathcal{S}}$. In fact, $\mathcal{P}_A^{pos}$ can be implemented without any

access to $\mathcal{P}_{\mathcal{S}}$, since it is known that the non-zero entries are on the diagonal, while $\mathcal{P}_A^{val}$ can be implemented with a single access to $\mathcal{P}_{\mathcal{S}}$, assuming that M and N are known: by definition we have $\mathcal{P}_{\mathcal{S}}|j,0\rangle = |j,0\rangle$ if $j \notin \mathcal{S}$ and $\mathcal{P}_{\mathcal{S}}|j,0\rangle = |j,1\rangle$ if $j \in \mathcal{S}$, thus it is sufficient to apply on the ancilla system the transformation $|0\rangle \mapsto |\alpha\rangle$ and $|1\rangle \mapsto |\beta\rangle$, where the quantum register contains a binary representation of the numbers α and β, to implement $\mathcal{P}_A^{val}$.

Now notice that A is a matrix having condition number $\kappa = \frac{\alpha}{\beta} = \sqrt{\frac{N-M}{M}} \in \Theta\Big(\sqrt{\frac{N}{M}}\Big)$ assuming $M \leq N/2$. Moreover, A^{-1} is also diagonal, with entries

$$(A^{-1})_{j,j} = \begin{cases} \sqrt{\frac{N}{N-M}} & \text{if } j \notin \mathcal{S} \\ \sqrt{\frac{N}{M}} & \text{if } j \in \mathcal{S}\,. \end{cases} \tag{14}$$

Solving the QLS for the known-term vector $|\mathbf{b}\rangle = |\mathbf{1}_N\rangle = \frac{1}{\sqrt{N}}\sum_{j=1}^{N}|j\rangle$ yields

$$|\mathbf{x}\rangle = \frac{A^{-1}|\mathbf{1}_N\rangle}{||A^{-1}|\mathbf{1}_N\rangle||} \tag{15}$$

$$= \frac{1}{\sqrt{2N}}\left(\sqrt{\frac{N}{N-M}}\sum_{j\notin\mathcal{S}}|j\rangle + \sqrt{\frac{N}{M}}\sum_{j\in\mathcal{S}}|j\rangle\right) \tag{16}$$

$$\equiv \frac{1}{\sqrt{2}}\Big(|j\notin\mathcal{S}\rangle + |j\in\mathcal{S}\rangle\Big), \tag{17}$$

where in the last line we have introduced the normalised vectors $|j\notin\mathcal{S}\rangle := \frac{1}{\sqrt{N-M}}\sum_{j\notin\mathcal{S}}|j\rangle$ and $|j\in\mathcal{S}\rangle := \frac{1}{\sqrt{M}}\sum_{j\in\mathcal{S}}|j\rangle$. Measuring $|\mathbf{x}\rangle$ in the computational basis therefore solves the search problem with probability $1/2$.

Suppose now that exists a quantum algorithm $\mathcal{A}$ that solves the QLS problem exactly ($\varepsilon = 0$) for PD matrices and that queries $\mathfrak{o}(\kappa)$ times the oracle $\mathcal{P}_A$. Applying $\mathcal{A}$ to the diagonal matrix A and $|\mathbf{b}\rangle = |\mathbf{1}_N\rangle$ would then require only $\mathfrak{o}(\sqrt{N/M})$ calls to $\mathcal{P}_A$, and hence to $\mathcal{P}_{\mathcal{S}}$, in order to produce $|\mathbf{x}\rangle$. This means that using $\mathcal{A}$ we can solve an unstructured search problem using $\mathfrak{o}(\sqrt{N/M})$ queries to $\mathcal{P}_{\mathcal{S}}$, violating the optimality of Grover search.

Next, suppose that the algorithm $\mathcal{A}$ is an approximate PD-QLS solver, i.e. that it produces a state $\rho_{\mathbf{x}}$ such that $\big|\big|\,\rho_{\mathbf{x}} - |\mathbf{x}\rangle\langle\mathbf{x}|\,\big|\big|_{\mathrm{Tr}} \leq \varepsilon$ for some constant $\varepsilon < 1/2$. Apply a projective measurement to $\rho_{\mathbf{x}}$ that projects it on the space spanned by $\{|j\rangle\}_{j\in\mathcal{S}}$ with probability p and projects it onto the orthogonal subspace with probability $q = 1-p$; in the ideal case ($\varepsilon = 0$) we would have $p = q = 1/2$. By the operational definition of the trace distance, the probability distribution (p,q) must be at most ε-distinguishable from $(1/2,1/2)$, i.e. $\max\{|p-1/2|, |q-1/2|\} \leq \varepsilon$. Thus, the success probability is a constant $p \geq 1/2 - \varepsilon > 0$.

Finally, this argument can be extended to any constant precision $\varepsilon < 1$. It is sufficient to define a new diagonal matrix $\widehat{A}$ by changing the values α and β in Eq. (13), so that the probability $\widehat{p}$ of finding an element $j \in \mathcal{S}$ when measuring $|\widehat{\mathbf{x}}\rangle = |\widehat{A}^{-1}\mathbf{1}_N\rangle$ satisfies $\widehat{p} > 1-\varepsilon$.

□

Notice that in the proof the vector $|\mathbf{b}\rangle = |\mathbf{1}_N\rangle$ is fixed and easy to produce and hence the access model for $|\mathbf{b}\rangle$ plays no role in our reduction. We also remark that this proof can be straightforwardly modified to prove that the operation of quantum matrix-vector multiplication (i.e., obtaining a state proportional to $A|\mathbf{b}\rangle$) must also have a linear cost in κ. Moreover, since a sparse oracle access $\mathcal{P}_A$ allows to efficiently implement also a matrix-block encoding $\mathcal{U}_A$ [5], the same reduction immediately rules out oracular algorithms that use $\mathfrak{o}(\kappa)$ accesses to $\mathcal{U}_A$ (for $\kappa \leq \sqrt{N}$). Finally, a simple argument shows that a $\Omega(\kappa)$ lower bound holds also for the $\mathcal{U}_{\mathbf{b}}$-query complexity: an initial small difference between two input states $\mathbf{b}$ and $\mathbf{b}'$ can be magnified κ-fold in the corresponding outputs $\big|A^{-1}\mathbf{b}\big\rangle$ and $\big|A^{-1}\mathbf{b}'\big\rangle$, which is impossible unless one uses at least κ accesses to $\mathcal{U}_{\mathbf{b}}$ [38].

4 Method based on low-degree polynomial approximations of A^{-1}

We start this section giving a few details on how to use the Quantum Signal Processing (QSP) method to implement polynomial functions of a matrix that we can access through a matrix-

block-encoding (Section 4.1) and then we provide the explicit definition of the approximating polynomials of the inverse function (Section 4.2). Next (Section 4.3) we discuss two cases where we can implement a matrix-block-encoding of $B = I - \eta A$, as required to achieve a quadratic reduction in the degree of the approximating polynomials. Finally (Section 4.4) we discuss the cost of matrix-vector multiplication and summarise the cost of PD-QLS solving via this approach.

4.1 The quantum signal processing method

We employ QSP as a tool to implement $\mathcal{U}_{A^{-1}}$, a matrix-block-encoding of A^{-1}, assuming that we have access to a normalised matrix-block-encoding $\mathcal{U}_B$ of

$$B := I - \eta A \tag{18}$$

for some $\eta > 0$. We assume that the spectrum of B is contained in the interval $\mathcal{D}_B = [-1, 1-\eta/\kappa]$, where κ is an upper bound to the condition number of A. The QSP method can be stated, already specialised to our case of interest, as follows [25, Theorem 56].

Theorem 8. *Consider a (β, b, ϵ)-block-encoding $\mathcal{U}_B$ of a Hermitian matrix B and let $P(x)$ be a degree-ℓ real polynomial with $|P(x)| \leq 1/2$ for all $x \in [-1, 1]$. Then there is a quantum circuit $\mathcal{U}_{P(B/\beta)}$ which is a $(1, b+2, 4\ell\sqrt{\epsilon/\beta})$-block-encoding of $P(B/\beta)$, and requires ℓ applications of U_B and $U_B^\dagger$, a single application of controlled-U_B and $\mathcal{O}\big((n+b)\ell\big)$ elementary quantum gates. Moreover, the same result holds for polynomials satisfying $|P(x)| \leq 1$ if $P(x)$ has defined parity, i.e., $P(-x) = P(x)$ or $P(-x) = -P(x)$.*

Importantly, there are explicit classical algorithms that can efficiently compute a parametrization of $\mathcal{U}_{P(B/\beta)}$ for any polynomial P and then compile an explicit quantum circuit that implements it, see Refs. [39, 40] for the current state-of-the-art.

We now further motivate the need to choose $\beta = 1$, i.e., that the matrix-block-encoding of B is normalised; equivalently, the part proportional to the identity in the definition (18) must not be rescaled. We remind that, as presented in Section 1.3, our goal is to implement an polynomial $P(B/\beta)$ approximating A^{-1}, that is

$$P(B/\beta) \approx f(B) = \frac{1}{I - B} = \frac{1}{I - (I - \eta A)} = \frac{1}{\eta} A^{-1}. \tag{19}$$

Note, however, that in this expression actually we have $P(x) \approx \frac{1}{1-\beta x}$, a function that has a singularity in $x = 1/\beta \leq 1$. As a consequence of Bernstein's inequality [29] this function may have polynomial approximations with quadratically better degree only when the singularity is in $x = 1$, i.e. when we have $\beta = 1$.

4.2 Polynomial approximation of $1/(1-x)$

In this section we show analytical polynomial approximations of the function $f(x) = 1/(1-x)$ so that we can use the QSP method to implement it for the matrix $B = I - \eta A$ as in Eq. (18). To keep the notation simple we assume $\eta = 1$ and that the spectrum of A is contained in $\mathcal{D}_A = \left[\frac{1}{\kappa}, 2\right]$, while we can account for any value $\eta < 1$ simply by rescaling κ to κ/η. Consequently, it is only necessary for the polynomial $P(x)$ to be a good approximation of our target function in the domain $\mathcal{D}_B = \left[-1, 1-\frac{1}{\kappa}\right]$.

Our starting point will be the polynomial $\hat{\mathcal{T}}_{\ell,\kappa}(x)$, a shifted and rescaled version of $\mathcal{T}_\ell(x)$, the ℓ-degree Chebyshev polynomial of the first kind,

$$\hat{\mathcal{T}}_{\ell,\kappa}(x) := \frac{\mathcal{T}_\ell\left(\frac{x+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)}{\mathcal{T}_\ell\left(\frac{1+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)}, \tag{20}$$

which is the solution of the following minimax problem (see Ref. [3, Theorem 6.25]):

$$\hat{\mathcal{T}}_{\ell,\kappa}(x) = \underset{\substack{P\in\mathbb{R}_\ell,\\ P(1)=1}}{\operatorname{argmin}} \max_{x\in[-1,1-\frac{1}{\kappa}]} \big|P(x)\big|. \tag{21}$$

Chebyschev polynomials satisfy the property that $|\mathcal{T}_\ell(x)| \leq 1$ for all $\ell \in \mathbb{N}$ and all $x \in [-1,+1]$, while $\mathcal{T}_\ell(1+\delta) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$ for $0 \leq \delta \leq 1/6$, see e.g. [12, Lemma 13]. Using the changes of variables

$$y(x) := \frac{x + \frac{1}{2\kappa}}{1 - \frac{1}{2\kappa}} \qquad \delta := \frac{1 + \frac{1}{2\kappa}}{1 - \frac{1}{2\kappa}} - 1 = \frac{1}{\kappa - 1/2} \tag{22}$$

we can rewrite the definition in Eq. (20) as

$$\hat{\mathcal{T}}_{\ell,\kappa}(x) = \frac{\mathcal{T}_\ell\big(y(x)\big)}{\mathcal{T}_\ell\big(1+\delta\big)}\,. \tag{23}$$

Then, the numerator satisfies $\big|\mathcal{T}_\ell\big(y(x)\big)\big| \leq 1$ for all $x \in \mathcal{D}_B = [-1, 1 - \frac{1}{\kappa}]$, while the denominator is $\mathcal{T}_\ell\big(1+\delta\big) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$. This means that it is sufficient to choose $\ell \geq \frac{1}{\sqrt{\delta}}\log\big(\frac{2}{\epsilon}\big)$, i.e. $\ell \in \Theta\big(\sqrt{\kappa}\log(1/\epsilon)\big)$, to obtain $\big|\hat{\mathcal{T}}_{\ell,\kappa}(x)\big| \leq \epsilon$ on the interval $\mathcal{D}_B$.

We then use the following $(2\ell-1)$-degree polynomial as our approximation of $f(x) = \frac{1}{1-x}$:

$$P_{2\ell-1,\kappa}(x) := \frac{1}{1-x}\left[1 - \hat{\mathcal{T}}_{\ell,\kappa}(x)\right]^2. \tag{24}$$

To see that $P_{2\ell-1,\kappa}(x)$ is indeed a polynomial, note that $[1 - \hat{\mathcal{T}}_{\ell,\kappa}(x)]$ has a twofold root in $x = 1$, thus is exactly divisible by $1 - x$ and moreover $P_{2\ell-1,\kappa}(1) = 0$. This last property is useful because it allows us to implement the pseudo-inverse A^+ for a singular matrix A; i.e., in the case in which A has some eigenvalues that are equal to 0 (equivalently, $B = I - A$ has eigenvalues equal to 1) these will be mapped to 0 and if moreover all the non-zero eigenvalues are separated from zero by a gap $1/\kappa$, the the polynomial in Eq. (24) is a close approximation of the mapping $(1-\lambda) \mapsto 1/\lambda$ for all $\lambda \neq 0$ in the spectrum of A. We choose the degree $\ell \in \Theta\big(\sqrt{\kappa}\log(\kappa/\varepsilon)\big)$ in such a way that $\big|\hat{\mathcal{T}}_{\ell,\kappa}(x)\big| \leq \varepsilon/(3\kappa)$ for all $x \in \mathcal{D}_B$ and thus we obtain from Eq. (24)

$$\left|P_{2\ell-1,\kappa}(x) - \frac{1}{1-x}\right| \;\leq\; \kappa\left(2\,\frac{\varepsilon}{3\kappa} + \frac{\varepsilon^2}{9\kappa^2}\right) \;\leq\; \varepsilon \qquad \forall x \in \mathcal{D}_B\,, \tag{25}$$

that is, we have an ε-close polynomial approximation on the interval $\mathcal{D}_B = [-1, 1 - \frac{1}{\kappa}]$. This directly implies $\big|\big|P_{2\ell-1,\kappa}(B) - A^{-1}\big|\big| \leq \varepsilon$ in operator norm.

Finally, the polynomial in Eq. (24) has to be normalised so that it becomes compatible with the QSP method. Therefore we define

$$\hat{P}_{2\ell-1,\kappa}(x) := \frac{P_{2\ell-1,\kappa}(x)}{K}\,, \qquad \text{where } K := 2\max_{x\in[-1,+1]}|P_{2\ell-1}(x)|\,. \tag{26}$$

With this definition we have $|\hat{P}_{2\ell-1}(x)| \leq \frac{1}{2}$ for $x \in [-1,+1]$, as required. The normalisation constant satisfies $K \in \Theta(\kappa)$ for $\ell \in \Omega(\sqrt{\kappa})$, see Appendix B for the proof of this bound. In conclusion, the QSP method allows us to implement a $(K, b+2, \varepsilon)$-matrix-block-encoding of A^{-1}, where b is the number of ancilla qubits required in the block-encoding of B.

4.3 Implementing normalised matrix-block-encodings of $B = I$ - ηA

In this subsection we show two methods that, under different assumptions, allow us to efficiently implement a normalised matrix-block-encoding of $B = I - \eta\, A$. Preliminarily, we remark that this is a non-trivial task, as we argue with the following three considerations.

First, any Hermitian matrix $M \in \mathbb{C}^{N\times N}$ satisfying $||M|| \leq 1$ can be implemented as a normalised sub-block of a unitary, since an explicit construction is given by [28]

$$\mathcal{U}_M = \begin{pmatrix} M & -\sqrt{I - M^2} \\ \sqrt{I - M^2} & M \end{pmatrix}. \tag{27}$$

Implementing the circuit corresponding to $\mathcal{U}_M$ in general requires up to $\mathcal{O}(N^2)$ elementary quantum gates [42] and is thus inefficient; however, efficient constructions are possible in specialised cases.

Second, standard quantum methods allow us to efficiently implement *sub-normalised* matrix-block-encodings of B. As a first example, Childs' quantum walk operator uses $\mathcal{O}(1)$ accesses to a sparse-matrix oracle $\mathcal{P}_B$ to implement $\mathcal{U}_M$ for $M = B/d$, where d is the column-sparsity of B [5]. A second example is to assume that we have access to $\mathcal{U}_A$, a block-encoding of A with $||A|| \leq 1$, and then use the LCU lemma [26] to implement a linear combination of I and $-\mathcal{U}_A$, yielding a normalised block-encoding of $p\,I - (1-p)A \equiv p\,B$, for some $p \in (0,1)$. However, any "black-box" method that aims at amplifying a block-encoding of B/β (with $\beta > 1$) to a normalised block-encoding of B is in general inefficient. This can be proven, for instance, applying the lower bound in Ref. [25, Theorem 73] to the function $f(x) = \beta\, x$.

Third, it is currently an open problem whether it is possible or not to implement a normalised matrix-block-encoding $\mathcal{U}_M$ given access to a sparse-matrix oracle $\mathcal{P}_M$ with $||M|| \leq 1$. In absence of general results of this kind, we then turn to developing specialised methods to efficiently implementing a normalised block-encoding $\mathcal{U}_B$, with $B = I - \eta\, A$, in the cases where (i) A is diagonally dominant or (ii) A is the sum of positive semi-definite local Hamiltonians.

4.3.1 Diagonally-dominant coefficient matrix

In this section, we implement a normalised block-encoding of $B = I - A$ employing the method described in Ref. [25, Lemma 47], which we report here in Lemma 9. Our construction requires the preparation of some families of states $\{|\psi_i\rangle\}_i, \{|\phi_j\rangle\}_j$ that are well-defined only when A is diagonally-dominant, while attempts at extending the method to $B = I - \eta\, A$ for non-diagonally-dominant PD matrices results in non-normalisable states for any $\eta > 0$. We also remark that the problem of solving linear systems involving diagonally dominant PD matrices (which includes the noteworthy class of Laplacian matrices [43]) is well studied in classical linear algebra: for these classes of matrices there are classical algorithms that can solve a linear system substantially faster than what is possible for more general matrices [44].

Lemma 9. *Suppose that we have access to two "state preparation" unitaries U_L and U_R (left and right) acting on $a + s$ qubits such that*

$$U_L : |0^a\rangle\,|i\rangle \;\mapsto\; |\psi_i\rangle \tag{28}$$

$$U_R : |0^a\rangle\,|j\rangle \mapsto |\phi_j\rangle\,, \tag{29}$$

for any $i, j \in \{1, \ldots, 2^s\}$ and for some families of states $\{|\psi_i\rangle\}_i$ and $\{|\phi_j\rangle\}_j$. Then, it is immediate to see that $U_L^\dagger U_R$ is a $(1, a, 0)$-matrix-block-encoding of the Gram matrix H such that $H_{ij} = \langle\psi_i|\phi_j\rangle$.

Let A be a Hermitian d-sparse diagonally-dominant matrix, i.e. $\sum_{j\neq i} |A_{ij}| \leq A_{ii} \leq 1$ for all i. By Gershgorin theorem [41], diagonal dominance of a Hermitian matrix is sufficient to guarantee positive semi-definiteness, i.e. $\lambda_{\min}(A) \geq 0$, and both $\lambda_{\min}(A) = 0$ and $\lambda_{\min}(A) \neq 0$ are possible[7]. Consider then the states

$$|\psi_i\rangle := \sum_{l\in \mathrm{supp}(A_i)} \sqrt{\delta_{il} - A_{il}}\,|l\rangle \;+\; \sqrt{r_i}\,|N+1\rangle \tag{30}$$

$$|\phi_j\rangle := \sum_{k\in \mathrm{supp}(A_j)} \sqrt{\delta_{jk} - A^*_{jk}}\,|k\rangle \;+\; \sqrt{r_j}\,|N+1\rangle \tag{31}$$

where $\mathrm{supp}(A_i)$ are the position of the (at most) d non-zero entries of the column vector A_i and the value $0 \leq r_i \leq 1$ can be computed so that the states are normalised, since we have

$$|r_i| = 1 - \sum_{l\in \mathrm{supp}(A_i)} \left|\sqrt{\delta_{il} - A_{il}}\right|^2 = A_{ii} - \sum_{l\neq i} \left|A_{il}\right| \geq 0 \tag{32}$$

where we have used $A_{ii} \leq 1$ and the diagonal dominance of A. We then define the following state-preparation unitaries:

$$U_L : |0^a\rangle\,|i\rangle \;\mapsto\; |0^b\rangle\,|i\rangle\,|\psi_i^*\rangle \tag{33}$$

$$U_R : |0^a\rangle\,|j\rangle \;\mapsto\; |0^b\rangle\,|\phi_j\rangle\,|j\rangle\,, \tag{34}$$

[7] If A is *strictly* diagonally dominant we have $\lambda_{\min}(A) \geq \min_i \left\{A_{ii} - \sum_{j\neq i} |A_{ij}|\right\} > 0$ and then the condition number is immediately bounded by $\kappa(A) \leq 1/\lambda_{\min}(A)$ when $||A|| \leq 1$.

for certain numbers a and b of ancilla qubits initialised in $|0\rangle$, and where $|\psi_i^*\rangle$ is the complex conjugate of the state $|\psi_i\rangle$ w.r.t. the computational basis. Then, $U_L^\dagger U_R$ is a normalised encoding of the matrix $B = I - A$, as one can verify using $A_{ji}^* = A_{ij}$:

$$B_{ij} = \langle 0^b, i, \psi_i^* | 0^b, \phi_j, j \rangle = \underbrace{\sqrt{\delta_{ji} - A_{ji}^*}}_{\langle i|\phi_j\rangle} \underbrace{\sqrt{\delta_{ij} - A_{ij}}}_{\langle \psi_i^*|j\rangle} = \delta_{ij} - A_{ij} \ . \tag{35}$$

The quantum circuit U_L can be implemented efficiently (and similarly for U_R), as we now show. We use d calls to $\mathcal{P}_A^{pos}$ to recover the values $j_\nu \equiv j(i,\nu)$ for $\nu \in \{1,\dots,d\}$, i.e., the positions of all the (potentially) non-zero entries of A_i. This corresponds to implementing the following isometry (i.e., a unitary circuit plus the possibility of adding ancillas):

$$|i\rangle \overset{d\times\mathcal{P}_A^{pos}}{\mapsto} |i\rangle\,|j_1,\dots,j_d\rangle\,. \tag{36}$$

Next, we use d calls to $\mathcal{P}_A^{val}$ to recover all the values $A_{i,j}$, i.e.:

$$(36) \overset{d\times\mathcal{P}_A^{val}}{\mapsto} |i\rangle\,|j_1,\dots,j_d\rangle\,|A_{ij_1},\dots,A_{ij_d}\rangle\,. \tag{37}$$

We then use reversible (classical) computation to calculate the numerical values of all the amplitudes $\boldsymbol{\psi}^{(i)} := (\sqrt{-A_{ij_1}}^*,\dots,\sqrt{1-A_{ii}}^*,\dots,\sqrt{-A_{ij_d}}^*,\sqrt{r_i}^*)^T$:

$$(37) \overset{\text{compute}}{\mapsto} |i\rangle\,|j_1,\dots,j_d\rangle\,|A_{ij_1},\dots,A_{ij_d}\rangle\,|\boldsymbol{\psi}^{(i)}\rangle\,. \tag{38}$$

Then, we use a general state preparation quantum circuit which, given a classical description of the amplitudes of a $(d+1)$-dimensional quantum state, effectively prepares the corresponding state:

$$(38) \overset{\text{prepare}}{\mapsto} |i\rangle\,|j_1,\dots,j_d\rangle\,|A_{ij_1},\dots,A_{ij_d}\rangle\,|\boldsymbol{\psi}^{(i)}\rangle \sum_{\nu=1}^{d+1} \psi_\nu^{(i)}\,|\nu\rangle\,. \tag{39}$$

Next, we use a single call to $\mathcal{P}_A^{pos}$ in quantum superposition to map $|i,\nu\rangle \mapsto |i, j(i,\nu)\rangle = |i, j_\nu\rangle$ and thus we obtain

$$(39) \overset{\mathcal{P}_A^{pos}}{\mapsto} |i\rangle\,|j_1,\dots,j_d\rangle\,|A_{ij_1},\dots,A_{ij_d}\rangle\,|\boldsymbol{\psi}^{(i)}\rangle \sum_{\nu=1}^{d+1} \psi_\nu^{(i)}\,|j_\nu\rangle \tag{40}$$

and now note that, adopting the definition $j(i,d+1) := N+1$, on the right-hand side we have obtained $\sum_{\nu=1}^{d+1} \psi_\nu^{(i)}\,|j_\nu\rangle = |\psi_i^*\rangle$, the complex conjugate (w.r.t. the computational basis) of the state defined in Eq. (30). Finally, we "uncompute" the intermediate registers $|j_1,\dots,j_d\rangle$, $|A_{ij_1},\dots,A_{ij_d}\rangle$ and $|\boldsymbol{\psi}^{(i)}\rangle$, mapping them to $|0^b\rangle$ (where b is the number of left-over ancillas), performing the steps (36) to (39) in reverse. With a final swapping of $|i\rangle$ and $|0^b\rangle$, we have implemented the circuit U_L given in Eq. (33).

We can now estimate the query and gate cost of implementing U_L (and the cost of implementing U_R is the same). Going through the derivation, we see that $4d+1$ oracle calls to $\mathcal{P}_A = (\mathcal{P}_A^{pos}, \mathcal{P}_A^{val})$ are required, that is the query complexity is $Q[\mathcal{P}_A] \in \mathcal{O}(d)$. Regarding gate complexity, step (38) requires $\mathcal{O}\big(d\,\mathrm{poly}(p)\big)$ Toffoli gates (which are universal for classical reversible computation) to perform the computation up to p digits of precision; moreover, step (39) can be performed using $\mathcal{O}(d)$ control-nots and single-qubit rotations employing the methods of Ref. [42]. In conclusion, treating the number of digits of precision as a constant and the single-qubit rotations as exact, both the query and the gate complexities are in $\mathcal{O}(d)$ and we thus obtain the following result.

Proposition 10 (Normalised matrix-block-encoding, diagonally-dominant case)**.** *Suppose that we have access to a d-sparse diagonally-dominant PD matrix $A \in \mathbb{C}^{N\times N}$ via $\mathcal{P}_A$ as in Definition 4. Then it is possible to implement a normalised block-encoding of $B = I - A$ with $\mathcal{O}(d)$ query and gate complexity, assuming exact single-qubit rotations and that all arithmetic operations are performed with a constant number of digits of precision.*

We finally remark that, in some cases, it might be possible to implement U_L and U_R with a query and gate complexity in $\mathcal{O}(\sqrt{d})$ instead of linearly in d. First, we may assume that we have directly access to an oracle $\mathcal{P}_\psi$ that directly returns the amplitudes $\psi_\nu^{(i)}$ (including the value $\sqrt{r_i}$), instead of needing to compute these vales from the non-zero entries of A_i. Second, one can use an algorithm that generalises Grover search to prepare the state $|\psi_i^*\rangle$ using $\mathcal{O}(\sqrt{d})$ accesses to $\mathcal{P}_\psi$ [45], and an even more efficient implementation can be realised using the method of Ref. [46], which avoids synthesising arithmetical operations and brings about an improvement of two orders of magnitude over prior works for realistic levels of precision.

4.3.2 Sum of positive semi-definite Hamiltonians

We now consider the case where $A \in \mathbb{C}^{N\times N}$ is given by the sum of positive semi-definite Hamiltonian terms; i.e., we consider the case

$$A = \sum_{j=1}^{J} H_{(j)} \qquad \forall j \; H_{(j)} \text{ is positive semi-definite,} \tag{41}$$

which is similar to the case presented in Section 5, but here we allow the Hamiltonian terms to have eigenvalue zero. We assume that the number J of Hamiltonian terms scales polynomially in $n = \lceil \log_2 N \rceil$ and that each Hamiltonian term $H_{(j)}$ is local, i.e., that it acts upon a small number of qubits; specifically, we require that the set $\mathcal{S}_j \subseteq \{1,\dots,n\}$ of qubits upon which H_j acts non-trivially satisfies $|\mathcal{S}_j| \leq s \in \mathcal{O}(\log n)$ for all j. Each $H_{(j)}$ can thus be expressed as[8]

$$H_{(j)} = h_{(j)} \otimes I_{\neg \mathcal{S}_j} \tag{43}$$

where $h_{(j)}$ is a positive-definite matrix of dimension $2^s \times 2^s$, which can be fully specified with $2^{\mathcal{O}(\log n)} = \mathcal{O}(\text{poly}\, n)$ parameters. We also impose $\left|\left|h_{(j)}\right|\right| \leq 2$ for all j.

Now we define $w_{(j)} := I - h_{(j)}$ and note that it is a small matrix, of at most $\mathcal{O}(\text{poly}\, n)$ size, with $\left|\left|w_{(j)}\right|\right| \leq 1$. Then, we can rapidly compute, with classical algorithms, a unitary extension $u_{(j)}$ for each $w_{(j)}$, each requiring only one ancilla qubit. Specifically, we define

$$u_{(j)} := \begin{pmatrix} w_{(j)} & -\sqrt{I - w_{(j)}^2} \\ \sqrt{I - w_{(j)}^2} & w_{(j)} \end{pmatrix} \tag{44}$$

and then we (implicitly) define $W_{(j)} \in \mathbb{C}^{N\times N}$ and the unitary $U_{(j)} \in \mathbb{C}^{2N\times 2N}$

$$W_{(j)} = w_{(j)} \otimes I_{\neg \mathcal{S}_j} \tag{45}$$
$$U_{(j)} = u_{(j)} \otimes I_{\neg \mathcal{S}_j} \tag{46}$$

where the interpretations are the same as in Eq. (43). Note that each $U_{(j)}$ can be efficiently compiled as a quantum circuit: it is sufficient to determine the gate decomposition of the $(s+1)$-qubit matrix $u_{(j)}$ and then embed the circuit in a $(n+1)$-qubit quantum register. We assume that the ancilla qubit used for the extension given in Eq. (44) is one and shared across all circuits $U_{(j)}$.

We then employ the LCU lemma [26] as follows. Given access to the controlled version of each circuit $U_{(j)}$, it is possible to efficiently implement the multi-controlled unitary

$$U_{\text{SELECT}} = \sum_{j=1}^{J} |j\rangle\langle j|_c \otimes U_{(j)} \tag{47}$$

[8] The correct expression for an operator $H_{(j)}$ acting on a set $\mathcal{S}_j \subseteq \{1,\dots,n\}$ of s qubits is

$$H_{(j)} = \sum_{\substack{a_1\cdots a_s \in \{0,1\}^s \\ b_1\cdots b_s \in \{0,1\}^s}} h^{(j)}_{a_1\cdots a_s, b_1\cdots b_s} \bigotimes_{r=1}^{n} \mathfrak{O}_r \qquad \text{with } \mathfrak{O}_r = \begin{cases} I = |0\rangle\langle 0| + |1\rangle\langle 1| & \text{if } r \notin \mathcal{S}_j \\ |a_r\rangle\langle b_r| & \text{if } r \in \mathcal{S}_j \end{cases}. \tag{42}$$

where the subscript c denotes the control register. Then, defining a unitary Had that acts as $\text{Had}\,|0\rangle_c = \sum_{j=1}^{J} \frac{1}{\sqrt{J}} |j\rangle_c$, we obtain:

$$\left(\langle 0|_c \,\text{Had}^\dagger \otimes I \right) U_{\text{SELECT}} \left(\text{Had}\,|0\rangle_c \otimes I \right) = \sum_{j=1}^{J} \frac{1}{J} U_{(j)} \tag{48}$$

and further post-selecting to the "top-left" corner of each $U_{(j)}$, according to Eq. (44) we obtain:

$$\left(\langle 0| \otimes I \right) \sum_{j=1}^{J} \frac{1}{J} U_{(j)} \left(|0\rangle \otimes I \right) = \sum_{j=1}^{J} \frac{1}{J} W_{(j)} = \sum_{j=1}^{J} \frac{1}{J} \left(I - H_{(j)} \right) = I - \frac{1}{J} A\,. \tag{49}$$

In conclusion, $\mathcal{U}_B := (\text{Had}^\dagger \otimes I)\, U_{\text{SELECT}}\, (\text{Had} \otimes I)$ is a normalised matrix-block-encoding of $B = I - \eta A$, where the factor $\eta = 1/J$ scales, by assumption, polynomially in n. The gate complexity of $\mathcal{U}_B$ can be estimated as follows. Each $u_{(j)}$ requires $\mathcal{O}(2^{2s})$ elementary gates to be implemented [42] and thus the gate complexity of $U_{(j)}$ is also in $\mathcal{O}(2^{2s})$, assuming that two-qubit gates can be applied among arbitrary pairs of qubits. Then, U_{SELECT} has gate complexity scaling as the sum of the complexities of the individual $U_{(j)}$ [27] and is thus in $\mathcal{O}(J 2^{2s})$. The complexity of Had is $\mathcal{O}(\log J)$, a sub-leading additive term that can be neglected in the asymptotic gate complexity of $\mathcal{U}_B$. Hence we have the following result.

Proposition 11 (Normalised matrix-block-encoding, Sum-of-Hamiltonians case)**.** *Suppose that we have an explicit classical description of positive semi-definite matrices $h_{(j)} \in \mathbb{C}^{2^s \times 2^s}$ with $j \in \{1, \ldots, J\}$ and consider the positive semi-definite Hamiltonian terms $H_{(j)}$ each obtained by applying $h_{(j)}$ to a subset of qubits $\mathcal{S}_j \subseteq \{1, \ldots, n\}$, as given in Eq. (43). Consider then a coefficient matrix $A \in \mathbb{C}^{2^n \times 2^n}$ as given in Eq. (41). Then it is possible to implement a normalised block-encoding of $B = I - \frac{1}{J} A$ with a gate complexity in $\mathcal{O}(J 2^s)$, assuming exact single-qubit rotations.*

4.4 From matrix inversion to solving the quantum linear system problem

Suppose now that we have a matrix-block-encoding of

$$\hat{P}_{2\ell-1,\kappa}(B) \approx \frac{1}{K} \frac{1}{I - B} = \frac{A^{-1}}{\eta K} \tag{50}$$

where $K \in \Theta(\kappa/\eta)$ is the normalization factor of the matrix-block-encoding (recall the rescaling of κ to κ/η), upper bounded by $\mathcal{O}(\kappa/\eta)$ as we show in Appendix B. This is equivalent to say that we have implemented the unitary

$$\mathcal{U}_{A^{-1}} \approx \begin{pmatrix} A^{-1}/(\eta K) & * \\ * & * \end{pmatrix} \tag{51}$$

where the left-upper block corresponds to having a ancilla qubits in $|0^a\rangle$. Then, one can directly solve a QLS by applying the unitary quantum circuit $\mathcal{U}_{A^{-1}}$ to the vector $|0^a\rangle\,|\mathbf{b}\rangle$ and then post-select the outcome $|0^a\rangle$ on the ancilla system; however, post-selection might introduce large overheads, since we have

$$\mathcal{U}_{A^{-1}} : |0^a\rangle\,|\mathbf{b}\rangle \quad \mapsto \quad \frac{1}{\eta K} \quad |0^a\rangle\, A^{-1}\,|\mathbf{b}\rangle + \sqrt{1 - \frac{1}{\eta^2 K^2}}\,\left|\Psi^\perp\right\rangle \tag{52}$$

$$= \frac{\left|\left| A^{-1}\,|\mathbf{b}\rangle \right|\right|}{\eta K}\,|0^a\rangle\,\left|A^{-1}\mathbf{b}\right\rangle + \sqrt{1 - \frac{1}{\eta^2 K^2}}\,\left|\Psi^\perp\right\rangle \tag{53}$$

where $\left|\Psi^\perp\right\rangle$ is a state perpendicular to all states of the form $|0^a, \psi\rangle$. Therefore, the probability of successfully obtaining the state $\left|A^{-1}\mathbf{b}\right\rangle$ when post-selecting on the ancilla measurement is

$$p_{\text{succ}} = \frac{\left|\left| A^{-1}\,|\mathbf{b}\rangle \right|\right|^2}{\eta^2 K^2}\,. \tag{54}$$

A PD-QLS solver that prepares a matrix-encoding of A^{-1} and obtains $\left|A^{-1}\mathbf{b}\right\rangle$ via post-selection requires $\mathcal{O}(1/p_{\text{succ}})$ accesses to $\mathcal{U}_{\mathbf{b}}$ and $\mathcal{O}((2\ell-1)/p_{\text{succ}})$ accesses to $\mathcal{U}_B$. Recall, $2\ell-1$ is the degree of $\hat{P}_{2\ell-1,\kappa}(B)$ and $\ell \in \Theta\Big(\sqrt{\frac{\kappa}{\eta}}\log\frac{\kappa}{\eta\varepsilon}\Big)$. The query complexities can be quadratically improved to $\mathcal{O}(1/\sqrt{p_{\text{succ}}})$ and $\mathcal{O}((2\ell-1)/\sqrt{p_{\text{succ}}})$, respectively, using amplitude amplification [30]. Having implemented an approximate matrix-encoding of $\widetilde{A}^{-1}$ that is ε-close to A^{-1} (using Definition 5), the output state $|\widetilde{A}^{-1}\mathbf{b}\rangle = \widetilde{A}^{-1}|\mathbf{b}\rangle / ||\widetilde{A}^{-1}|\mathbf{b}\rangle||$ satisfies [5, Proposition 9]

$$\Big|\Big|\left|\widetilde{A}^{-1}\mathbf{b}\right\rangle - \left|A^{-1}\mathbf{b}\right\rangle\Big|\Big| \leq 4\,\varepsilon\,. \tag{55}$$

The same inequality holds in trace distance because of (5) and it is true both when using post-selection and when using amplitude amplification. In conclusion, we have the following results.

Proposition 12 (Complexity of the PD-QLS solver)**.** *Suppose that we have access to a $(1,b,0)$-matrix-block-encoding $\mathcal{U}_B$ of $B = I - \eta A$, where $\eta \in (0,1]$ and $A \in \mathbb{C}^{N\times N}$ is a PD matrix with eigenvalues contained in the interval $\mathcal{D}_A = \left[\frac{1}{\kappa},2\right]$ for some known value $\kappa > 1$; see e.g. Proposition 10 and Proposition 11 for explicit constructions.*

First, using the QSP method of Theorem 8 and the polynomial approximation given in Eq. (24), it is possible to implement a $(K,b+2,\varepsilon)$-matrix-block-encoding of A^{-1}, where $K \in \Theta(\kappa/\eta)$, and the method has a query complexity

$$Q[\mathcal{U}_B] \in \mathcal{O}\left(\sqrt{\frac{\kappa}{\eta}}\log\frac{\kappa}{\eta\,\varepsilon}\right), \tag{56}$$

where $Q[\mathcal{U}_B]$ denotes the number of accesses to $\mathcal{U}_B$. Moreover, the algorithm is gate-efficient, i.e. it requires $\mathcal{O}\big(\text{poly}(n)\,Q[\mathcal{U}_B]\big)$ extra elementary quantum gates.

Second, suppose that we want to solve a PD-QLS as in Definition 2, where access to A is given (indirectly) by $\mathcal{U}_B$ and access to $\boldsymbol{b}$ via a state preparation oracle $\mathcal{U}_{\boldsymbol{b}}$ as in Definition 3. Then, the QLS can be solved up to precision $\mathcal{O}(\varepsilon)$ using the $(K,b+2,\varepsilon)$-matrix-block-encoding of A^{-1} and employing amplitude amplification to perform matrix-vector multiplication with constant success probability. The total query complexities, in terms of accesses to $\mathcal{U}_B$ and $\mathcal{U}_{\boldsymbol{b}}$, are

$$Q[\mathcal{U}_{\boldsymbol{b}}] \in \mathcal{O}\left(\frac{\kappa}{||A^{-1}|\boldsymbol{b}\rangle||}\right) \tag{57}$$

$$Q[\mathcal{U}_B] \in \mathcal{O}\left(\sqrt{\frac{\kappa}{\eta}}\frac{\kappa}{||A^{-1}|\boldsymbol{b}\rangle||}\log\frac{\kappa}{\eta\,\varepsilon}\right) \tag{58}$$

and the algorithm is gate efficient, that is, the gate complexity is in $\mathcal{O}\big(Q\,\text{poly}(\log Q,\log N)\big)$. A quadratic speed-up in κ (up to polylogarithmic factors) is achieved over general QLS solvers when $\Big|\Big|A^{-1}|\boldsymbol{b}\rangle\Big|\Big| \in \mathcal{O}(\kappa)$.

We now proceed to a worst-case, average case, and best-case scenario analysis of a PD-QLS solver as given in the previous Proposition.

Worst-case scenario: In the worst case we have $\Big|\Big|A^{-1}|\mathbf{b}\rangle\Big|\Big| \in \mathcal{O}(1)$ and consequently the post-selection success probability is $p_{\text{succ}} \in \Omega(1/\kappa^2)$. One can in alternative use $\mathcal{O}(\kappa)$ rounds of amplitude amplification to reach a constant success probability. Using amplitude amplification, the overall query complexity is in $\mathcal{O}(\kappa^{3/2})$, which is an improvement compared to the $\mathcal{O}(\kappa^2)$ runtime achieved by the HHL algorithm, but still falls shorts of the $\widetilde{\mathcal{O}}(\kappa)$ runtime that can be achieved using more advanced methods such as Variable-Time Amplitude Amplification (VTAA) [4] or eigenpath transversal [7].

Average-case scenario: We now look at the distribution of runtimes that arises when using a randomly chosen vector $|\mathbf{b}\rangle$. As observed in the work by Subaşı and Somma [47, Section III.B] one could model the eigenvalues of a positive-definite matrix A as uniformly distributed over the interval $[1/\kappa,1]$ while if $|\mathbf{b}\rangle$ is chosen from the outcome of a random quantum circuit its amplitudes are sampled according to a Porter-Thomas distribution; then, $\Big|\Big|A^{-1}|\mathbf{b}\rangle\Big|\Big| \in \mathcal{O}(\sqrt{\kappa})$ almost surely

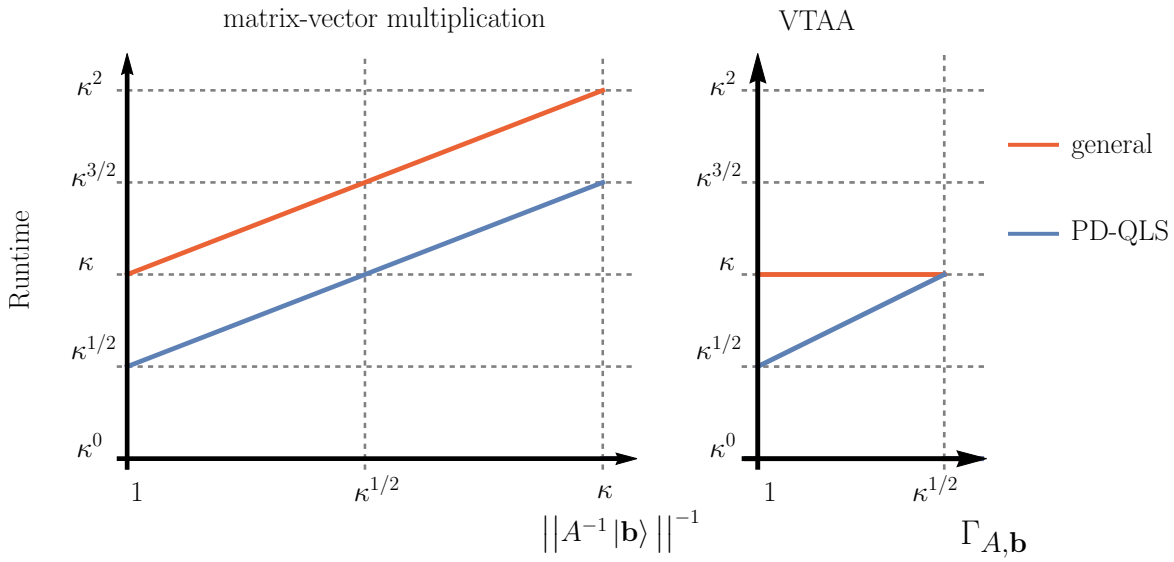

Figure 3: Bi-logarithmic plots of the asymptotic runtimes of general QLS solvers and of our PD-QLS solvers. The plot on the left represents the scaling of the query complexity for methods based on direct matrix-vector multiplication (together with amplitude amplification) in terms of the variable $||A^{-1}|\mathbf{b}\rangle||^{-1} \in [1, \kappa]$. The plot on the right represents the scaling of the query complexity for methods based on VTAA in terms of the variable $\Gamma_{A,\mathbf{b}} := \sqrt{\kappa}\,\frac{||A^{-1/2}|\mathbf{b}\rangle||}{||A^{-1}|\mathbf{b}\rangle||} \in [1, \sqrt{\kappa}]$. In both plots we ignore poly-logarithmic multiplicative factors and we express all the variables in units of powers of κ, assuming $\kappa \gg 1$ as constant. See the main text for details.

in the regime $1 \ll \kappa \ll N$. This implies that a randomly sampled PD-QLS problem (according to the specified distribution) can be solved almost surely with a query complexity in $\mathcal{O}(\kappa)$, employing amplitude amplification in the post-selection step. This method then matches (actually, improves by a polylog(κ) factor) the asymptotic runtime of more sophisticated methods (such as those that employ VTAA or adiabatic evolution) when considering "typical" instances of PD-QLS.

Best-case scenario: The largest value that $||A^{-1}|\mathbf{b}\rangle||$ can reach is κ. In such case the post-selection probability is constant and the overall query complexity is in $\mathcal{O}(\sqrt{\kappa})$, even without employing amplitude-amplification. This is a quadratic improvement for PD-QLS solving over competing methods working for indefinite QLS, since implementing a block-encoding of A^{-1} for indefinite matrices already requires $\mathcal{O}(\kappa)$ oracle calls to $\mathcal{U}_A$ [5]. We note these best-case problems almost never occur under the probabilistic model described before, but real-world problems have intrinsic structure that could make them depart from the Porter-Thomas distribution and thus have $||A^{-1}|\mathbf{b}\rangle|| \gg \sqrt{\kappa}$: for instance, this is the case if $|\mathbf{b}\rangle$ has constant overlap with the eigenvector relative to the largest eigenvalue of A^{-1}. Note, finally, that it is not required that we know in advance how large the success probability is, since by definition $|0^a\rangle$ heralds the success.

4.5 Optimization using Variable-Time Amplitude Amplification

In Appendix C we show how to use VTAA to obtain a PD-QLS solver having improved asymptotic query complexities. We proceed as in Ref. [5, Section 5] and Ref. [19, Section 3]: first we reformulate our algorithm as a variable-stopping-time quantum algorithm and afterwards we apply the VTAA optimisation to improve its runtime. There is, however, a technical hurdle to overcome: all previous VTAA-based QLS solvers use a phase estimation subroutine having a $\mathcal{O}(\kappa)$ runtime; its use would then preclude us from achieving a runtime sub-linear in κ. The main new idea we introduce is to replace phase estimation with efficient "windowing functions" whose implementation via QSP requires only $\widetilde{\mathcal{O}}(\sqrt{\kappa})$ accesses to $\mathcal{U}_B$.

More precisely, in Ref. [5] the so-called Gapped Phase Estimation (GPE) method is introduced, with the purpose of reliably selecting eigenvalues of A that are larger than some value δ. For these eigenvalues it is possible to implement an approximate inverse at a reduced cost, scaling as $\mathcal{O}(1/\delta)$ instead of $\mathcal{O}(\kappa)$. Then, a sequence of increasingly small values of δ is considered, until $\delta \leq 1/\kappa$, and VTAA is employed to enhance the success probability. Since GPE has a query complexity in $\Omega(1/\delta)$ and the required precision is $\delta \leq 1/\kappa$, its complexity is in $\Omega(\kappa)$.

Instead, we introduce a "windowing function" $W_{\epsilon,\delta}(\lambda)$ to select eigenvalues of $B = I - \eta A$ that satisfy $\lambda \in [-1+2\delta, 1-2\delta]$ and to reject eigenvalues $\lambda \in [-1, -1+\delta] \cup [1-\delta, +1]$, except for a small error probability ϵ. Thus, $W_{\epsilon,\delta}(x)$ is a polynomial ϵ-close to 1 in the center of the interval $[-1, +1]$ and ϵ-close to 0 near the edges of the interval, with a steep fall around the points $\pm(1-1.5\delta)$. The intervals where the function derivative is large (of order $1/\delta$) are very close to the extrema of the interval $[-1, +1]$. According to Bernstein's inequality [29] it is not prohibited that a windowing function could be implemented with a polynomial having a degree $\ell \in \widetilde{O}(1/\sqrt{\delta})$, a quadratically smaller degree compared to case where the large derivative is near the center of the interval. In Appendix C we then show, with an explicit construction, that windowing polynomial with degree $\ell \in \widetilde{O}(1/\sqrt{\delta})$ indeed can be implemented. We defer to the Appendix for further details.

The end result is summarized in the following Proposition.

Proposition 13 (Complexity of PD-QLS with VTAA)**.** *Consider a PD-QLS where we have access to a normalised matrix-encoding of $B = I - \eta A$ and to a state preparation unitary for $\boldsymbol{b}$. Then, there is a VTAA-based solver having target precision ε, constant success probability, and query complexities given by*

$$Q[\mathcal{U}_{\boldsymbol{b}}] \in \mathcal{O}\left(\sqrt{\log(\kappa)} + \frac{\kappa}{||A^{-1}\,|\boldsymbol{b}\rangle||}\right) \tag{59}$$

$$Q[\mathcal{U}_B] \in \mathcal{O}\left(\sqrt{\frac{\kappa}{\eta}}\,\Gamma_{A,\boldsymbol{b}}\,\mathrm{polylog}(\kappa, \tilde{\epsilon}^{-1}, \eta^{-1})\right) \tag{60}$$

$$\text{with} \quad \Gamma_{A,\boldsymbol{b}} := \sqrt{\kappa}\,\frac{\left|\left|A^{-1/2}\,|\boldsymbol{b}\rangle\right|\right|}{||A^{-1}\,|\boldsymbol{b}\rangle||}, \qquad \tilde{\epsilon} \in \mathcal{O}\left(\frac{\varepsilon}{\kappa\sqrt{\log\kappa}}\right) \tag{61}$$

$$\text{and} \quad \mathrm{polylog}(\kappa, \tilde{\epsilon}^{-1}, \eta^{-1}) = \log^2(\kappa)\log^{7/4}(\tilde{\epsilon}^{-1})\log^{3/2}(\eta^{-1}). \tag{62}$$

Moreover, the algorithm is gate efficient.

Now we discuss the runtime of this VTAA PD-QLS solver.

- We always have $Q[\mathcal{U}_B] \geq Q[\mathcal{U}_{\mathbf{b}}]$, thus the $\mathcal{U}_B$-complexity is the dominant factor.
- Compared to Proposition 12, the $\mathcal{U}_{\mathbf{b}}$ query complexity increases here only by an additive $\sqrt{\log(\kappa)}$ factor, while the $\mathcal{U}_B$ complexity typically (i.e., for almost all values of $\Gamma_{A,\mathbf{b}}$) has a polynomial improvement. To prove it, note that $\left|\left|A^{-1/2}\,|\mathbf{b}\rangle\right|\right|^2 = \langle\mathbf{b}|\,A^{-1}\,|\mathbf{b}\rangle \leq \kappa$ and thus $\Gamma_{A,\mathbf{b}} = \sqrt{\kappa}\,\frac{\left|\left|A^{-1/2}|\mathbf{b}\rangle\right|\right|}{||A^{-1}|\mathbf{b}\rangle||} \leq \frac{\kappa}{||A^{-1}|\mathbf{b}\rangle||}$.
- Compared to the general VTAA-based QLS solver of Ref. [5], our PD-QLS solver has a polynomial speed-up for almost all values of $\Gamma_{A,\mathbf{b}}$, see the right plot in Figure 3. This is a consequence of Lemma 20, where we prove that $\Gamma_{A,\mathbf{b}} \in [1, \sqrt{\kappa}]$.

5 Method based on a quadratic reduction of the condition number via matrix decomposition

In this section we start giving some preliminary considerations on the approach that we are going to present (Section 5.1) and then give a formal statement of the Sum-QLS problem that we solve (Section 5.2). Next, we describe a classical pre-processing step that quadratically reduces the condition number (Section 5.3) and then present the quantum algorithm solving the pseudo-inversion problem that originates from the preconditioning (Section 5.4). Finally, we estimate the gate complexity of the resulting Sum-QLS solver (Section 5.5).

5.1 General considerations

We present now the general features of any algorithm that solves PD-QLS exploiting a decomposition of the form $A = LL^\dagger$, as already summarised in Section 1.4. Note that such L exists for any PD matrix A and that it may not be unique, especially if we allow L to be non-square. The key property is that any L such that $A = LL^\dagger$ satisfies $\kappa_{\text{eff}}(L) = \sqrt{\kappa(A)}$, hence a system of the form $L^\dagger \mathbf{x} = \mathbf{b}'$ (for any $\mathbf{b}'$) is quadratically better conditioned than the original system $A\mathbf{x} = \mathbf{b}$.

The method we introduce is based on finding matrices $L \in \mathbb{C}^{N\times M}$ and $L^g \in \mathbb{C}^{M\times N}$, with $M > N$, such that the decomposition $A = LL^\dagger$ holds and L^g satisfies $LL^g = I$, i.e. it is a right pseudo-inverse; moreover, L^g is rectangular with more rows than columns and thus $L^g L \neq I$, i.e. L^g cannot be a left pseudo-inverse. We make then the following observations.

1. $L^g LL^\dagger \mathbf{x} = L^g \mathbf{b}$ is a linear system equivalent to the original one, having the vector $\mathbf{x} = A^{-1}\mathbf{b}$ as the unique solution, but with no guarantee that the condition number of $L^g LL^\dagger$ is small.
2. $L^\dagger \mathbf{x} = L^g \mathbf{b}$ is a over-constrained linear system that may be inequivalent to the original one (since $L^g L \neq I$) and typically has no proper solution $\mathbf{x}$.
3. Finding $\operatorname{argmin}_{\mathbf{x}} \left|\left| L^\dagger \mathbf{x} - L^g \mathbf{b} \right|\right|$ is a problem equivalent to the original system. The unique solution is $\mathbf{x} = (L^\dagger)^+ L^g \mathbf{b}$, therefore using[9] $(L^\dagger)^+ = (LL^\dagger)^{-1} L = A^{-1} L$ and $LL^g = I$ we get the required result $\mathbf{x} = A^{-1} LL^g \mathbf{b} = A^{-1}\mathbf{b}$.

The goal is thus to convert the linear system $A\mathbf{x} = \mathbf{b}$ into the linear regression problem $\operatorname{argmin}_{\mathbf{x}} \left|\left| L^\dagger \mathbf{x} - L^g \mathbf{b} \right|\right|$, having solution $\mathbf{x} = (L^\dagger)^+ L^g \mathbf{b}$. This is a non-trivial task, as we need, given access to A via sparse-matrix oracle or via some succinct description, to construct a suitable access to $L^\dagger$ and, moreover, given access to $\mathbf{b}$, to construct a suitable access to $\mathbf{b}' := L^g \mathbf{b}$. The latter requirement seems particularly worrisome, since it involves a pseudo-inversion of the exponentially large matrix L. We show, however, that for the Sum-QLS problem, whereby A is provided as a sum of local PD terms, one can find a suitable L^g for which a compact classical description can be efficiently computed.

We also remark that a pseudo-inversion problem can be interpreted as a regular matrix inversion on the subspace where the matrix $L^\dagger$ is full-rank; thus, solving pseudo-inversion entails a larger runtime compared to solving a standard QLS, since the appropriate subspace has to be selected via projection or via amplitude amplification. More formally, the operator $(L^\dagger)^+ : \mathbb{C}^M \to \mathbb{C}^N$ (with $M > N$) has rank equal to N and thus we have the orthogonal decomposition

$$\mathbb{C}^M = \operatorname{supp}\big((L^\dagger)^+\big) + \ker\big((L^\dagger)^+\big) \qquad \begin{cases} \dim \operatorname{supp}\big((L^\dagger)^+\big) = N \\ \dim \ker\big((L^\dagger)^+\big) = M - N \end{cases} \tag{63}$$

where the support is by definition the subspace orthogonal to the kernel. Then, calling Π and $\Pi^\perp = I - \Pi$ the orthogonal projectors on the support and on the kernel of $(L^\dagger)^+$, respectively[10], we obtain the identity

$$(L^\dagger)^+ \left|\mathbf{b}'\right\rangle = (L^\dagger)^+ \Pi \left|\mathbf{b}'\right\rangle + (L^\dagger)^+ \Pi^\perp \left|\mathbf{b}'\right\rangle = (L^\dagger)^+ \Pi \left|\mathbf{b}'\right\rangle \tag{64}$$

since by definition we have $(L^\dagger)^+ \Pi^\perp = 0$. It is then evident that only the component $\Pi \left|\mathbf{b}'\right\rangle$, which is in general a sub-normalised quantum state, plays a role in the pseudo-inversion algorithm, while the orthogonal component $\Pi^\perp \left|\mathbf{b}'\right\rangle$ can be arbitrary. Therefore, any quantum pseudo-inversion algorithms implicitly requires the amplification of the $\Pi \left|\mathbf{b}'\right\rangle$ component, which therefore entails a gate complexity in $\mathcal{O}(1/\sqrt{\gamma})$, for some known lower bound $\sqrt{\gamma} \leq \left|\left|\Pi \left|\mathbf{b}'\right\rangle\right|\right|$.

5.2 Problem statement

In the Sum-QLS we assume that the coefficient matrix $A \in \mathbb{C}^{N\times N}$ is given by an *explicit classical description*, rather than via oracular access. This allows us to evade the lower bounds given in

[9] By assumption A is invertible, hence $L^\dagger \in \mathbb{C}^{M\times N}$ is full-rank, and thus using the SVD $L^\dagger = W\Sigma^\dagger V^\dagger$ we get $(LL^\dagger)^{-1} L = (V\Sigma\Sigma^\dagger V^\dagger)^{-1} V\Sigma W^\dagger = V(\Sigma^\dagger)^+ W^\dagger = (L^\dagger)^+$.

[10] Using the identity $(L^\dagger)^+ = A^{-1}L$ we obtain $\operatorname{supp}((L^\dagger)^+) = \operatorname{supp}(L)$ and moreover $\ker((L^\dagger)^+) = \ker(L)$.

Proposition 6, since those bounds are formulated for relativising (i.e. oracular) algorithms. We assume, specifically, that A has the form [16]

$$A = \sum_{j=1}^{J} H_{(j)} \qquad \forall j \; H_{(j)} \text{ is positive definite.} \tag{65}$$

Here we impose that each $H_{(j)}$ is strictly positive definite (rather than semi-definite) because of a technical condition that will become clear later; in essence, the expression in Eq. (91) can diverge if any $H_{(j)}$ is singular, resulting in an infinite runtime. Each term $H_{(j)}$ is a local Hamiltonian, i.e. it can be expressed as [see also Eq. (42) in the footnote]

$$H_{(j)} = h_{(j)} \otimes I_{\neg \mathcal{S}_j} \tag{66}$$

where each $h_{(j)}$ is an operator acting on a subset $\mathcal{S}_j \subseteq \{1, \dots, n\}$ of at most s qubits, corresponding to a matrix of size at most $2^s \times 2^s$. We assume that J, the number of Hamiltonian terms, and s are "small", i.e. we take $J \in \mathcal{O}(\text{poly}\, n)$ and $s \in \mathcal{O}(\log n)$, and that we have a complete classical description of each operator $h_{(j)}$ and of each subset $\mathcal{S}_j$. The matrix A is fully specified with at most $J2^{2s}$ real parameters and with Jn boolean values (defining the sets $\mathcal{S}_j$), i.e. the number of parameters is $J2^{2s} + Jn \in \mathcal{O}(\text{poly}\, n)$.

We require moreover that the known-term vector $\mathbf{b}$ is sparse, containing at most $d_{\mathbf{b}}$ non-zero entries in the computational basis, where $d_{\mathbf{b}}$ also scales polynomially in n. This implies that a preparation circuit $\mathcal{U}_{\mathbf{b}}$ can be given as an explicit small quantum circuit. This leads to the following definition for the Sum-QLS problem.

Definition 14 (Sum-of-Hamiltonians Quantum Linear System)**.** *A Sum-QLS problem is a PD-QLS as in Definition 2 with the following restrictions. The coefficient matrix $A \in \mathbb{C}^{N \times N}$, for $N = 2^n$, is provided as the sum of PD Hamiltonian terms $A = \sum_{j=1}^{J} H_{(j)}$, where each $H_{(j)}$ acts on at most s qubits; each $H_{(j)}$ is fully specified as a PD matrix $h_{(j)}$ of size $2^{s_j} \times 2^{s_j}$ with $s_j \leq s$, together with the set $\mathcal{S}_j$ of s_j qubits on which $H_{(j)}$ acts upon. The vector $\boldsymbol{b} \in \mathbb{C}^N$ is $d_{\boldsymbol{b}}$-sparse and the value and position of each of the $d_{\boldsymbol{b}}$ non-zero entries is provided.*

In Appendix D we show that the Sum-QLS problem is BQP-hard, by adapting a proof given in HHL [1]. That is, we show that any polynomial-time quantum computation (in the BQP class) can be re-formulated as a Sum-QLS problem for some artfully constructed coefficient matrix A and known-term vector $\mathbf{b}$; therefore no polynomial-time classical probabilistic algorithm (in the BPP class) can solve the Sum-QLS problem[11] (unless BPP = BQP).

5.3 Classical pre-processing step

In this section we describe the classical pre-processing step, providing a quadratic improvement of the condition number. We decompose each matrix $h_{(j)}$ as

$$h_{(j)} = l_{(j)} l_{(j)}^{\dagger}\,, \tag{67}$$

which can be accomplished for example via the Cholesky decomposition [31], in which case $l_{(j)}$ is a lower triangular matrix. Notice that each matrix $h_{(j)}$ is a small matrix of size $2^s \times 2^s \in \mathcal{O}(\text{poly}\, n)$ and thus the Cholesky decomposition can be performed numerically on a classical computer using $\mathcal{O}(\text{poly}\, n)$ operations; specifically, Cholesky factorisation of a $m \times m$ matrix requires $m^3/3$ arithmetic operations, $m^3/6$ additions and $m^3/6$ multiplications [31]. The total number of Hamiltonian terms is $J \in \mathcal{O}(\text{poly}\, n)$, implying that the total runtime for performing the Cholesky decomposition for all Hamiltonian terms is $\mathcal{O}(2^{3s} J)$, which also is polynomial in n under our assumptions.

We save the Cholesky decompositions $l_{(j)}$ in a classical memory, storing $\mathcal{O}(J2^2 s) = \mathcal{O}(\text{poly}\, n)$ complex values, for later use. These decompositions implicitly define the operators

$$L_{(j)} = l_{(j)} \otimes I_{\neg \mathcal{S}_j} \tag{68}$$

[11] In this context, a classical probabilistic algorithm "solves" a QLS problem if it outputs a value $n \in \{1, \dots, N\}$ with probability approximately equal to $|x_n|^2$, the square of the n-th amplitude of the quantum state $|\mathbf{x}\rangle = \left|A^{-1}\mathbf{b}\right\rangle$.

where each $L_{(j)}$ is of size $2^n \times 2^n$ and where the interpretation of this equation is the same as in Eq. (66). We now introduce the rectangular matrix $L \in \mathbb{C}^{N \times JN}$, with $N = 2^n$, given by

$$L := \left(L_{(1)} \;\middle|\; \cdots \;\middle|\; L_{(J)} \right) \tag{69}$$

and thus we have the required decomposition

$$LL^\dagger = \sum_{j=1}^{J} L_{(j)} L_{(j)}^\dagger = \sum_{j=1}^{J} H_{(j)} = A\,. \tag{70}$$

We can efficiently implement quantum circuits $\mathcal{P}_L$ ($\mathcal{P}_{L^\dagger}$) that provide sparse-matrix access to L ($L^\dagger$). To this end, it is sufficient to convert the classical random access memory that stores the positions and values of the entries of L ($L^\dagger$) into a qRAM [35]. The scheme presented in Ref. [48, Section 6.3.5] implements this qRAM with a gate complexity in $\mathcal{O}(nJ2^{2s})$ and a circuit depth in $\mathcal{O}\big(\log(J2^{2s})\big) = \mathcal{O}\big(s + \log J\big)$.

Since $h_{(j)}$ is by assumption non-singular, each $l_{(j)}$ is a non-singular lower-triangular matrix and its inverse $l_{(j)}^{-1}$ is an upper-triangular matrix which we can efficiently compute and store in a classical memory using a polynomial amount of space. These matrices implicitly define operators $L_{(j)}^{-1} = l_{(j)}^{-1} \otimes I_{\neg \mathcal{S}_j}$ such that $L_{(j)} L_{(j)}^{-1} = I$. We can then define the matrix

$$L^g := \frac{1}{J} \left(\begin{array}{c} L_{(1)}^{-1} \\ \hline \vdots \\ \hline L_{(J)}^{-1} \end{array} \right). \tag{71}$$

which is a generalised right pseudo-inverse of L, i.e. L^g satisfies the equation

$$L\, L^g = \frac{1}{J} \sum_{j=1}^{J} I = I \tag{72}$$

and thus using $(L^\dagger)^+ = A^{-1} L$ we get the required relation $(L^\dagger)^+ L^g = A^{-1} L\, L^g = A^{-1}$. Using a qRAM the gate complexity of $\mathcal{P}_{L^g}$ is equal to that of $\mathcal{P}_L$ and is thus in $\mathcal{O}(nJ2^{2s})$.

We finally introduce the quantum state

$$\big|\mathbf{b}'\big\rangle := |L^g\, \mathbf{b}\rangle\,. \tag{73}$$

By assumption, $\mathbf{b}$ is sparse and has $d_{\mathbf{b}} \in \mathcal{O}(\text{poly}\, n)$ non-zero entries, hence the vector $\mathbf{b}'$ has sparsity $d_{\mathbf{b}'} \leq d_{\mathbf{b}} J 2^s \in \mathcal{O}(\text{poly}\, n)$. It is then possible efficiently classically compute the positions and the values of all the non-zero entries of $\mathbf{b}'$, with a gate complexity in $\mathcal{O}(n\, d_{\mathbf{b}'})$, and thus also compute the normalisation factor $\big|\big|\mathbf{b}'\big|\big|$, with a gate complexity in $\mathcal{O}(d_{\mathbf{b}'})$. Using the method described in Ref. [49], we can then efficiently compile a quantum circuit $\mathcal{U}_{\mathbf{b}'}$ that prepares $\big|\mathbf{b}'\big\rangle$ and has a gate complexity in $\mathcal{O}(n\, d_{\mathbf{b}'}) = \mathcal{O}(n\, d_{\mathbf{b}} J 2^s)$, assuming that all single qubit rotations are performed exactly.

Employing the classical pre-processing described up to now, we can efficiently implement quantum circuits $\mathcal{P}_{L^\dagger}$ and $\mathcal{U}_{\mathbf{b}'}$ that act as a sparse access to $L^\dagger$ and state preparation circuit for $\mathbf{b}'$, respectively; importantly, the gate complexities of these unitaries are independent from κ. We thus have at hand the necessary tools to implement the quantum pseudo-inversion algorithm that we present in the upcoming Section 5.4.

5.4 Efficient pseudo-inversion quantum algorithm

In this section we look into a quantum algorithm for the linear regression problem

$$\underset{\mathbf{x}}{\text{argmin}} \,\big|\big|\, L^\dagger \mathbf{x} - \mathbf{b}' \,\big|\big| \tag{74}$$

having solution $|\mathbf{x}\rangle = \big|(L^\dagger)^+ \mathbf{b}'\big\rangle = \big|A^{-1}\mathbf{b}\big\rangle$. We then employ the quantum pseudo-inversion algorithm of [19, Corollary 31] which we report here for completeness.

Proposition 15 (Complexity of pseudo-inverse state preparation)**.** *Suppose $\widetilde{\kappa} \geq 2$, $\mathcal{L} \in \mathbb{C}^{N\times N}$ is a Hermitian matrix whose non-zero eigenvalues are contained in the domain $[-1,-1/\widetilde{\kappa}]\cup[1/\widetilde{\kappa},1]$, and ε is the target precision. Assume that we have access to $\mathcal{U}_{\mathcal{L}}$, an (α,a,δ)-matrix-block-encoding of $\mathcal{L}$ with $\delta \in \mathfrak{o}\left(\varepsilon/(\widetilde{\kappa}^2\log^3\frac{\widetilde{\kappa}}{\varepsilon})\right)$ and $a \in \Omega(\log N)$, and to a state preparation oracle $\mathcal{U}_{\boldsymbol{v}}$ for a vector $\boldsymbol{v}$ such that $||\Pi_{\mathcal{L}}|\boldsymbol{v}\rangle|| \geq \sqrt{\gamma}$, where $\Pi_{\mathcal{L}}$ is the orthogonal projector onto the support of $\mathcal{L}^+$ and γ is a known positive parameter. Then, there is a (VTAA-based) quantum algorithm that produces a state ε-close to $|\mathcal{L}^+\boldsymbol{v}\rangle$ and has:*

$$Q\left[\mathcal{U}_{\mathcal{L}}\right] \in \mathcal{O}\left(\frac{\alpha}{\sqrt{\gamma}}\widetilde{\kappa}\log^3(\widetilde{\kappa})\log^2(1/\varepsilon)\right) \tag{75}$$

$$Q\left[\mathcal{U}_{\boldsymbol{v}}\right] \in \mathcal{O}\left(\frac{1}{\sqrt{\gamma}}\widetilde{\kappa}\log(\widetilde{\kappa})\right) \tag{76}$$

where $Q[\mathcal{U}_{\mathcal{L}}]$ and $Q[\mathcal{U}_{\boldsymbol{b}}]$ are the query complexity in terms of access to $\mathcal{U}_{\mathcal{L}}$ and $\mathcal{U}_{\boldsymbol{b}}$. The algorithm is gate-efficient, only requiring $\mathcal{O}\big(a\,Q\left[\mathcal{U}_{\mathcal{L}}\right]\big)$ extra elementary gates.

We will apply Proposition 15 using as coefficient matrix $\mathcal{L}$ the Hermitian extension of $L^\dagger$ and $\widetilde{\kappa} \equiv \sqrt{\kappa}$. That is, we consider the Hermitian matrix $\mathcal{L} \in \mathbb{C}^{(J+1)N\times(J+1)N}$ and vector $\mathbf{v} \in \mathbb{C}^{(J+1)N}$

$$\mathcal{L} = \begin{pmatrix} 0 & L^\dagger \\ L & 0 \end{pmatrix} \quad \mathbf{v} = \begin{pmatrix} \mathbf{b}' \\ 0 \end{pmatrix} \tag{77}$$

which, after pseudo-inversion, results in the vector

$$\mathbf{x} = \mathcal{L}^+\mathbf{v} = \begin{pmatrix} 0 & L^+ \\ (L^\dagger)^+ & 0 \end{pmatrix}\begin{pmatrix} \mathbf{b}' \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ (L^\dagger)^+\,\mathbf{b}' \end{pmatrix} \tag{78}$$

which encodes the quantum state $|J\rangle \otimes \left|(L^\dagger)^+\,\mathbf{b}'\right\rangle$ and the solution is obtained discarding the $(J+1)$-level ancilla system in the state $|J\rangle$.

5.5 Runtime estimation

In this section we look into methods for estimating of the parameters α, κ and γ (i.e., the normalisation of the block-encoding of $L^\dagger$, condition number, and overlap with the support space) that determine the runtime of the pseudo-inversion algorithm of Proposition 15, and thus determine the overall complexity of the Sum-QLS solver.

First, we can implement sparse-matrix-accesses $\mathcal{P}_L$ and $\mathcal{P}_{L^\dagger}$ using the information stored in a qRAM, as previously explained. Childs' walk operator [5] then allows to realise a block-encoding $\mathcal{U}_{\mathcal{L}}$ of $\mathcal{L}$ using $\mathcal{O}(1)$ accesses to $\mathcal{P}_L$ and $\mathcal{P}_{L^\dagger}$. Note that $\mathcal{U}_{\mathcal{L}}$ is also a block-encoding of $L^\dagger$ (after a swap of the position of the block). The normalisation factor of this block-encoding is $\alpha = J2^s \in \mathcal{O}(\text{poly}\,n)$, equal to the sparsity of $\mathcal{L}$.

Second, we can explicitly bound the condition number of A as follows. Positive-definiteness of the Hamiltonian terms $H_{(j)}$ implies positive-definiteness of A and, moreover, the smallest and largest eigenvalues of A satisfy $\lambda_{\min}(A) \geq \sum_{j=1}^J \lambda_{\min}(h_{(j)})$ and $\lambda_{\max}(A) \leq \sum_{j=1}^J \lambda_{\max}(h_{(j)})$. Since each $h_{(j)}$ can be efficiently diagonalised, this means that it is possible to classically compute these values, which then yield the explicit upper bound

$$\kappa(A) \leq \frac{\sum_{j=1}^J \lambda_{\max}(h_{(j)})}{\sum_{j=1}^J \lambda_{\min}(h_{(j)})} \equiv \kappa\,. \tag{79}$$

Tighter bounds to $\kappa(A)$ could also be obtained via more computationally intensive numerical methods, e.g. by first summing together groups of Hamiltonian terms and then diagonalizing each sum of Hamiltonians.

We now move on to lower-bounding the value of the overlap parameter

$$\left|\left|\,\Pi_{\mathcal{L}}\,|\mathbf{v}\rangle\,\right|\right| = \left|\left|\,\Pi_L\,|\mathbf{b}'\rangle\,\right|\right|, \tag{80}$$

where $\Pi_{\mathcal{L}}$ are the projectors on the supports of $\mathcal{L}^+$ and of $(L^\dagger)^+$, respectively, and note that the supports of L and $(L^\dagger)^+$ are equal. Using the identity $\Pi_L = L^+ L$ we thus obtain

$$\Pi_L = L^\dagger A^{-1} L = \sum_{i,j=1}^{J} |i\rangle\langle j| \otimes L^\dagger_{(i)} A^{-1} L_{(j)} \,. \tag{81}$$

Moreover, we have:

$$\left|\mathbf{b}'\right\rangle = \left|L^g\,\mathbf{b}\right\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=1}^{J} |j\rangle \otimes L^{-1}_{(j)} \left|\mathbf{b}\right\rangle \tag{82}$$

where the normalisation factor $\mathcal{N}$ is given by

$$\mathcal{N} = \sum_{j=1}^{J} \left|\left| L^{-1}_{(j)} \left|\mathbf{b}\right\rangle \right|\right|^2 = \sum_{j=1}^{J} \left\langle \mathbf{b}\right| H^{-1}_{(j)} \left|\mathbf{b}\right\rangle . \tag{83}$$

Then we can compute

$$\Pi_L \left|\mathbf{b}'\right\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{i,j=1}^{J} |i\rangle \otimes L^\dagger_{(i)} A^{-1} L_{(j)} L^{-1}_{(j)} \left|\mathbf{b}\right\rangle \tag{84}$$

$$= \frac{J}{\sqrt{\mathcal{N}}} \sum_{i=1}^{J} |i\rangle \otimes L^\dagger_{(i)} A^{-1} \left|\mathbf{b}\right\rangle \tag{85}$$

and finally we obtain

$$\left|\left|\Pi_L \left|\mathbf{b}'\right\rangle\right|\right|^{-1} = \frac{\sqrt{\mathcal{N}}}{J} \left[\left\langle \mathbf{b}\right| A^{-1} \textstyle\sum_{i=1}^{J} \left(L_{(i)} L^\dagger_{(i)}\right) A^{-1} \left|\mathbf{b}\right\rangle\right]^{-1/2} \tag{86}$$

$$= \frac{1}{J} \sqrt{\frac{\sum_{j=1}^{J} \left\langle \mathbf{b}\right| H^{-1}_{(j)} \left|\mathbf{b}\right\rangle}{\left\langle \mathbf{b}\right| A^{-1} \left|\mathbf{b}\right\rangle}} \,. \tag{87}$$

We now suppose that a value $\gamma > 0$ such that $||\Pi_{\mathcal{L}} |\mathbf{v}\rangle|| \geq \sqrt{\gamma}$ is known and we remind that the quantum psuedo-inversion algorithm has a runtime quasi-linear in $1/\sqrt{\gamma}$. We extensively comment on the values that γ can take in order to understand in which cases the Sum-QLS solver yields an advantage over competing methods.

1. We have $\left|\left|\Pi_L \left|\mathbf{b}'\right\rangle\right|\right| \leq 1$, and the inequality is saturated when $H_{(j)} = A/J$ for all j.

2. The bound $\sum_{j=1}^{J} \left\langle \mathbf{b}\right| H^{-1}_{(j)} \left|\mathbf{b}\right\rangle \leq J\,\lambda_*^{-1}$ holds, where $\lambda_* := \min_j \lambda_{\min}(H_{(j)})$. Assuming that $\lambda_* \in \Omega\big(\lambda_{\min}(A)/J\big)$, i.e. there is no Hamiltonian term having a minimum eigenvalue significantly smaller than the average minimum eigenvalue, we obtain:

$$\left|\left|\Pi_L \left|\mathbf{b}'\right\rangle\right|\right|^{-1} \in \mathcal{O}\left(\frac{1}{J} \frac{\sqrt{J\lambda_*^{-1}}}{\sqrt{\left\langle \mathbf{b}\right| A^{-1} \left|\mathbf{b}\right\rangle}}\right) = \mathcal{O}\left(\frac{\sqrt{\kappa(A)}}{\sqrt{\left\langle \mathbf{b}\right| A^{-1} \left|\mathbf{b}\right\rangle}}\right) . \tag{88}$$

3. The numerator in Eq. (87) can be explicitly calculated, while the denominator is in general difficult to compute[12]. However, assuming $||A|| \leq 1$, we have the bounds

$$\sqrt{\left\langle \mathbf{b}\right| A^{-1} \left|\mathbf{b}\right\rangle} = \left|\left| A^{-1/2} \left|\mathbf{b}\right\rangle \right|\right| \in \left[\,1, \sqrt{\kappa}\,\right] \tag{89}$$

4. Importantly, the expression $\left\langle \mathbf{b}\right| A^{-1} \left|\mathbf{b}\right\rangle$ appears at the denominator, so that a more "ill-conditioned" vector $\mathbf{b}$ results in a larger overlap and thus in a faster Sum-QLS solver.

[12] One could use techniques related to amplitude estimation to bound $\left|\left| A^{-1/2} \left|\mathbf{b}\right\rangle \right|\right|$, but this operation could be as difficult as solving the QLS in the first place.

5. As in the analysis of Section 4.4 we can study the runtime in an average-case scenario. For randomly chosen A and $\mathbf{b}$ (sampled according to suitable probability distributions) we have $\big|\big|A^{-1}\left|\mathbf{b}\right\rangle\big|\big| \in \Theta(\sqrt{\kappa(A)})$ almost surely; under the same assumptions, we also have $\big|\big|A^{-1/2}\left|\mathbf{b}\right\rangle\big|\big| \in \Theta(\kappa(A)^{1/4})$ almost surely. Inserting this estimation in Eq. (88) we have that

$$\big|\big|\Pi_L\left|\mathbf{b}'\right\rangle\big|\big|^{-1} \in \Theta\Big(\kappa(A)^{1/4}\Big) \tag{90}$$

holds almost surely. We conclude that for an average-case Sum-QLS problem the runtime is in $\widetilde{\mathcal{O}}(\sqrt{\kappa/\gamma}) = \widetilde{\mathcal{O}}(\kappa^{3/4})$, if $\sqrt{\gamma}$ is a tight lower bound for $\big|\big|\Pi_L\left|\mathbf{b}'\right\rangle\big|\big|$.

Summarising, we have the following result.

Proposition 16 (Complexity of the Sum-QLS solver)**.** *Consider the Sum-QLS problem with parameters N, J, s, κ and $d_{\boldsymbol{b}}$ as in Definition 14. There is a classical-quantum algorithm $\mathcal{A}$ solving the Sum-QLS that has the following features.*

The first part of $\mathcal{A}$ consists of an efficient classical pre-processing algorithm, which outputs a description of the quantum circuits implementing $\mathcal{U}_{L^\dagger}$ and $\mathcal{U}_{\boldsymbol{b}'}$; here $\mathcal{U}_{L^\dagger}$ is a $(2^s J, 1, 0)$-matrix-block-encoding of $L^\dagger$ [as given in Eq. (69)] with gate complexity in $\mathcal{O}(nJ2^{2s})$ and circuit depth in $\mathcal{O}\big(s + \log J\big)$; while $\mathcal{U}_{\boldsymbol{b}'}$ is a state preparation unitary for $\boldsymbol{b}'$ [as given in Eq. (71)] with gate complexity in $\mathcal{O}(n\, d_{\boldsymbol{b}}\, J2^s)$. By definition, $L^\dagger$ and $\boldsymbol{b}'$ satisfy $(L^\dagger)^+\boldsymbol{b}' = A^{-1}\boldsymbol{b}$.

The second part of $\mathcal{A}$ consists of using the quantum pseudo-inversion algorithm given in Proposition 15, using the unitaries $\mathcal{U}_{L^\dagger}$ and $\mathcal{U}_{\boldsymbol{b}'}$ as sub-routines, in order to produce a ε-close approximation of the ideal output $\left|\boldsymbol{x}\right\rangle = \left|A^{-1}\boldsymbol{b}\right\rangle$. This algorithm requires the knowledge of a value $\sqrt{\gamma}$ that satisfies $\sqrt{\gamma} \leq \big|\big|\Pi_L\boldsymbol{b}'\big|\big|$, where Π_L is the projector onto the support of L, equivalently:

$$\frac{1}{J^2}\frac{\sum_{j=1}^{J}\left\langle\boldsymbol{b}\right|H_{(j)}^{-1}\left|\boldsymbol{b}\right\rangle}{\left\langle\boldsymbol{b}\right|A^{-1}\left|\boldsymbol{b}\right\rangle} \leq \frac{1}{\gamma}\,. \tag{91}$$

Inserting the previously given expressions for the relevant parameters in Eqs. (75-76) results in

$$\text{gate complexity} \in \mathcal{O}\left(\left(n\,J2^{2s}\right)\frac{J2^s}{\sqrt{\gamma}}\,\kappa\log^3(\kappa)\log^2(1/\varepsilon) \;+\; \left(n\,d_{\boldsymbol{b}}J2^s\right)\frac{1}{\sqrt{\gamma}}\,\kappa\log(\kappa)\right) \tag{92}$$

$$= \mathcal{O}\left(\sqrt{\frac{\kappa}{\gamma}}\,\mathrm{poly}\Big(n, \log(\kappa/\varepsilon)\Big)\right) \tag{93}$$

assuming that $J, 2^s$ and $d_{\boldsymbol{b}}$ have polynomial dependence on $n = \log_2 N$. A quadratic speed-up in κ (up to polylogarithmic factors) is achieved over general QLS solvers when $\gamma \in \Omega(1)$.

We remark that the family of Sum-QLS instances where all the parameters in Proposition 16 scale polynomially (with the promise, in particular, that Eq. (91) holds for some $\gamma \in \mathcal{O}(\mathrm{poly}\, n)$), can be solved in polynomial time on a quantum computer, as was already shown in Ref. [16]. This means that the subset of problems having a polynomial scaling of the parameters, which we denote Sum-QLS$_{\mathrm{poly}}$, is contained in BQP. Moreover, the reduction in Appendix D can map polynomial-sized quantum circuits onto an instance of Sum-QLS$_{\mathrm{poly}}$, thus showing that Sum-QLS$_{\mathrm{poly}}$ is also BQP-hard. These two inclusion then show that Sum-QLS$_{\mathrm{poly}}$ is BQP-complete[13].

6 Discussion and outlook

In this work we have presented two algorithms aiming at solving QLS problems in the case where the coefficient matrix is positive definite and having (for certain problem instances) a runtime in $\mathcal{O}(\sqrt{\kappa})$, a quadratic improvement compared to what can be obtained using general QLS solvers. This improvement has the potential of greatly expanding the classes of problems where quantum computation can provide a quantum speed-up. For instance, the discretization of partial differential

[13] While the original QLS problem was proven to be BQP-complete already in Ref. [1], our contribution is to show that adding the constraint that A is the sum of positive-definite local Hamiltonians does not change the complexity class of the problem.

equations in D dimensions results in PD linear system with $\kappa \in \mathcal{O}(N^{2/D})$ [14] and thus having a runtime improvement from $\mathcal{O}(\kappa)$ to $\mathcal{O}(\sqrt{\kappa})$ is crucial to yield a quantum speed-up in the physically relevant cases $D = 2$ and $D = 3$. As a second example, it is possible estimate the hitting time of a Markov chain solving a QLS for the matrix $A = I - S$, where S is related to the discriminant matrix of the Markov chain [16]; since A is positive definite and decomposable as a sum of PD local Hamiltonian terms [16, Appendix A] our second algorithm could be applicable to this problem.

In the spirit of finding in the near future real-world applications of quantum algorithms, we note that there is considerable interest in the possibility of realising QLS solvers in Noisy Intermediate-scale Quantum (NISQ) devices [10, 11] and we argue that some of our results might be implementable in NISQ devices too. In particular, the crux of our first algorithm is to find a good polynomial approximation for A^{-1} with a degree in $\mathcal{O}(\sqrt{\kappa})$ and then implement it with the quantum signal processing method [28, 25]. The quadratic reduction in the degree of the polynomials renders their realisation more easily compatible with the next generation of quantum processors.

We note that many further improvements and extensions to our algorithms may be possible. Regarding the first algorithm (Section 4), it would be important to extend the classes of matrices for which a normalised matrix-block-encoding of $B = I - \eta\, A$ can be efficiently implemented to make the method more generally applicable. Regarding the second algorithm (Section 5) we note that the specific choice of the generalised pseudo-inverse L^g in the classical step results in a $\mathcal{O}(1/\sqrt{\gamma})$ multiplicative overhead in the runtime, where γ is given in Eq. (91). An open question is whether a different choice of the pseudo-inverse could improve, or eliminate altogether, this overhead. We also mention the possibility that the decomposition of A as a sum of local PD Hamiltonians could be computed on-the-fly by the solver, instead of being given as an external input. We note that if the sparsity pattern satisfies certain conditions (it is a *chordal* graph) a decomposition $A = LL^\dagger$ that does not increase the sparsity of A exists, and the characterisation given in Ref. [50, Theorem 2.6] could be employed to compute it.

We finally mention an open research idea that may be worth investigating. The eigenpath transversal method has been used in some new algorithms to solve the QLS problem with time complexity in $\widetilde{\mathcal{O}}(\kappa)$ [7, 8, 10, 11, 12]; this method is simpler than the Variable-Time Amplitude Amplification (VTAA) method and also results in (marginally) improved runtimes, however, it is not directly applicable to solve a pseudo-inversion problem. It would be interesting to find a way to adapt the eigenpath transversal method to make it work also in the case where the coefficient matrix is singular. As a by-product, it could replace the algorithm given in Ref. [19] as the subroutine used in our second algorithm to solve the pseudo-inversion problem, therefore making it more practical.

Acknowledgments

This work was supported by the Dutch Research Council (NWO/OCW), as part of the Quantum Software Consortium programme (project number 024.003.037). Some ideas present in the paper originated from discussions with Anirban Chowdhury while DO was visiting the Center for Quantum Information and Control (CQuIC). The authors acknowledge discussion with Markus Mieth, Anderas Spörl, and thank András Gilyén for reading the manuscript and for giving us several insightful comments and suggestions.

Appendices

A Proof of the query complexity lower bound

In this Appendix we give the proof of the query complexity lower bound presented in Section 3, which we present here in a more extended form.

Proposition 17 (Query complexity lower bound)**.** *Consider oracular quantum algorithms that solve the PD-QLS problem as presented in Definition 2 for different access models to A and $\boldsymbol{b}$. Namely, access to $\boldsymbol{b}$ is given via a state preparation oracle $\mathcal{U}_{\boldsymbol{b}}$ (Definition 3), while access to A is given either via a sparse-matrix oracle $\mathcal{P}_A$ (Definition 4) or via a matrix-block-encoding $\mathcal{U}_A$*

(Definition 5). Then, PD-QLS solving algorithms reaching a constant precision $\varepsilon \in \mathcal{O}(1)$ have query complexities $Q[\mathcal{U}_b], Q[\mathcal{U}_A], Q[\mathcal{P}_A]$ all in $\Omega\big(\min(\kappa, N)\big)$. More precisely, we have:

1. *$Q[\mathcal{U}_b] \in \Omega\big(\min(\kappa, N)\big)$, independently from the access model for A;*
2. *$Q[\mathcal{U}_A] \in \Omega\big(\min(\kappa, N)\big)$, independently from the access model for $\mathbf{b}$, when $\mathcal{U}_A$ is a normalised matrix-block-encoding;*
3. *$Q[\mathcal{P}_A] \in \Omega\big(\min(\kappa, N)\big)$, independently from the access model for $\mathbf{b}$, when A is a matrix with constant sparsity.*

In the main text we prove a weaker result using a reduction to the quantum search problem, which has a query complexity in $\Omega(\sqrt{N/M})$, where $M \in [N] := \{1, \dots, N\}$ is the number of marked elements; here, we use instead a reduction to a "promise majority" problem, which has a query complexity in $\Omega(N/M)$, where $M \in [N]$ is the margin of the majority. As a result, we can prove that solving a PD-QLS has linear scaling of the query complexity in the condition number for all $\kappa \in \mathcal{O}(N)$. To prove Proposition 17, we first introduce the PROMISEMAJORITY$_M$ problem as follows.

Definition 18 (PROMISEMAJORITY$_M$)**.** *Given a vector $y \in \{0,1\}^N$, a value $M \in [N]$ (we assume for simplicity that $N + M$ is even) and given the promise that we either have*

(Case 0) $y_i = 0$ for $N/2 + M/2$ of the entries
(Case 1) $y_i = 1$ for $N/2 + M/2$ of the entries

the PROMISEMAJORITY$_M$ problem consists in determining which of the two is the case.

We assume that we have access to y via a quantum oracle $\mathcal{P}_y$ that acts as $\mathcal{P}_y |i, z\rangle = |i, z \oplus y_i\rangle$ for all $i \in [N]$ and for $z \in \{0,1\}$. We also remind that the two-sided bounded-error quantum query complexity $\mathcal{Q}_2$ of a boolean function is defined as the minimum number of accesses to the input of the function (i.e., to $\mathcal{P}_y$) that are necessary to correctly output the value of the function with probability at least $2/3$, both for the positive and for the negative instances. Then we have the following Lemma:

Lemma 19. *The two-sided bounded-error quantum query complexity $\mathcal{Q}_2$ of PROMISEMAJORITY$_M$, in terms of accesses to $\mathcal{P}_y$, is $\mathcal{Q}_2(\text{PROMISEMAJORITY}_M) \in \Omega(N/M)$.*

Proof. This follows immediately from Ref. [51, Corollary 1.2]. □

We now show that (relativising) PD-QLS solving algorithms can be used to compute PROMISEMAJORITY$_M$; the lower bound on the query complexity of PROMISEMAJORITY$_M$ directly translates into a lower bound on the query complexity of the PD-QLS solvers. We will prove separately the three cases of Proposition 17, with each proof building upon the previous ones.

Proof. Case 1.
We assume that $y \in \{0,1\}^N$ is in the domain of PROMISEMAJORITY$_M$, i.e. y either contains exactly $N/2 + M/2$ zeros or $N/2 + M/2$ ones, and we define $\mathbf{b} \in \mathbb{C}^{N+1}$:

$$\begin{cases} b_i = (-1)^{y_i} & \text{for } i \in [N] \\ b_{N+1} = \sqrt{N+M} \end{cases} \tag{A.1}$$

where the value b_{N+1} is fixed to provide a "phase reference" and avoid ambiguity on the global sign. We have $|\mathbf{b}\rangle = \mathbf{b}/\sqrt{2N+M}$ and $|\mathbf{b}\rangle$ can be implemented by first preparing a state proportional to $(1, \dots, 1, \sqrt{N+M})$ and then applying the correct phases; this can be done with one oracle call to each of $\mathcal{P}_y$ and $\mathcal{P}_y^\dagger$, via the transformations

$$|i\rangle \overset{\text{ancilla}}{\mapsto} |i,0\rangle \overset{\mathcal{P}_y}{\mapsto} |i,y_i\rangle \overset{I\otimes Z}{\mapsto} (-1)^{y_i}|i,y_i\rangle \overset{\mathcal{P}_y^\dagger}{\mapsto} (-1)^{y_i}|i,0\rangle \overset{\text{discard}}{\mapsto} (-1)^{y_i}|i\rangle \tag{A.2}$$

and extended by linearity to superpositions. Next, we introduce the vector $\mathbf{1}_N := (1, \dots, 1)^T$ containing N ones and then define

$$K' \equiv K \oplus 0 := \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \oplus 0 \tag{A.3}$$

as a matrix of size $(N+1)\times(N+1)$, so that $K'^2 = K'$, and finally

$$A := I - (1-\epsilon)K' \tag{A.4}$$

where ϵ is a small parameter that we will define shortly. The matrix A can be used as a coefficient matrix for a PD-QLS solver since A is positive definite and $||A|| = 1$. Moreover, the condition number of A is exactly $\kappa(A) = 1/\epsilon$.

Next we have:

$$A^{-1} = I + \sum_{t=1}^{\infty} \left((1-\epsilon)K'\right)^t \tag{A.5}$$

$$= I + \frac{1-\epsilon}{\epsilon} K', \tag{A.6}$$

where the summation converges since $||(1-\epsilon)K|| < 1$. Let's apply A^{-1} to $\mathbf{b}$:

$$A^{-1}\mathbf{b} = \begin{cases} \mathbf{b} + \frac{1-\epsilon}{\epsilon}\frac{M}{N}\mathbf{1}'_N & \text{if } y \text{ has a majority of 0} \\ \mathbf{b} - \frac{1-\epsilon}{\epsilon}\frac{M}{N}\mathbf{1}'_N & \text{if } y \text{ has a majority of 1} \end{cases} \tag{A.7}$$

$$= \begin{cases} \mathbf{b} + \mathbf{1}'_N & \text{if } y \text{ has a majority of 0} \\ \mathbf{b} - \mathbf{1}'_N & \text{if } y \text{ has a majority of 1} \end{cases} \tag{A.8}$$

where we have introduced $\mathbf{1}'_N := (1,\dots,1,0)^T$ and we have chosen ϵ so that $\frac{1-\epsilon}{\epsilon}\frac{M}{N} = 1$, giving $\kappa(A) = \frac{1}{\epsilon} = \frac{N+M}{M}$. Introducing a boolean value $f = \textsc{PromiseMajority}_M(y)$, i.e. $f \in \{0,1\}$ is equal to the majority of y, we can rewrite the vector $A^{-1}\mathbf{b}$ entry-wise as

$$\begin{cases} [A^{-1}\mathbf{b}]_i & = (-1)^f \cdot 2 & \text{if } i \text{ is such that } y_i = f \\ [A^{-1}\mathbf{b}]_i & = 0 & \text{if } i \text{ is such that } y_i \neq f \\ [A^{-1}\mathbf{b}]_{N+1} & = \sqrt{N+M} \end{cases} \tag{A.9}$$

Then we have $\left|A^{-1}\mathbf{b}\right\rangle = A^{-1}\mathbf{b} / \left|\left|A^{-1}\mathbf{b}\right|\right|$, with

$$\left|\left|A^{-1}\mathbf{b}\right|\right|^2 = 2^2\,\frac{N+M}{2} + \sqrt{N+M}^{\,2} = 3(N+M) \tag{A.10}$$

so that we get

$$\left|A^{-1}\mathbf{b}\right\rangle = \sqrt{\frac{1}{3}}\left|N+1\right\rangle + (-1)^f\sqrt{\frac{2}{3}}\sum_{i:y_i=f}\sqrt{\frac{2}{N+M}}\left|i\right\rangle. \tag{A.11}$$

We then perform a projective measurement where one of the possible measurement outcomes is

$$\left|\text{“}+\text{”}\right\rangle := \sqrt{\frac{1}{2}}\left|N+1\right\rangle + \sqrt{\frac{1}{2}}\sum_{i=1}^{N}\frac{1}{\sqrt{N}}\left|i\right\rangle. \tag{A.12}$$

The cases $f=0$ and $f=1$ in Eq. (A.11) can be distinguished with constant advantage, since:

$$\left\langle\text{“}+\text{”}\right|A^{-1}\mathbf{b}\rangle = \sqrt{\frac{1}{6}} + (-1)^f\frac{N+M}{2}\sqrt{\frac{1}{3}}\sqrt{\frac{2}{N(N+M)}} \tag{A.13}$$

$$= \frac{1}{\sqrt{6}}\left(1 + (-1)^f\sqrt{1+M/N}\right). \tag{A.14}$$

Note that the two cases can still be distinguished with constant probability if we replace $|\mathbf{x}\rangle = \left|A^{-1}\mathbf{b}\right\rangle$ with any approximation $\rho_{\mathbf{x}}$ which is sufficiently close to it.

To summarise, suppose we have to solve a $\textsc{PromiseMajority}_M$ problem and that we can exploit as a subroutine an oracular quantum algorithm $\mathcal{A}$ that, given access to $\mathcal{U}_{\mathbf{b}}$, prepares the state $\left|A^{-1}\mathbf{b}\right\rangle$ with sufficiently high precision; suppose moreover that $\mathcal{A}$ has a query complexity $Q[\mathcal{U}_{\mathbf{b}}] =$

$g(\kappa)$, for some function $g : \mathbb{R}^+ \to \mathbb{N}$. Then, $\mathcal{A}$ can be used to prepare the state in Eq. (A.11) and solve PROMISEMAJORITY$_M$ with constant distinguishing advantage. The $\mathcal{P}_y$-query complexity of $\mathcal{A}$ is $Q[\mathcal{P}_y] = 2\,Q[\mathcal{U}_\mathbf{b}] = 2\,g\big(\kappa(A)\big) = 2\,g\big(\frac{N+M}{M}\big)$. The lower bound $\mathcal{Q}_2(\text{PROMISEMAJORITY}_M) \in \Omega(N/M)$ then directly implies $g(\kappa) \in \Omega\big(\min(\kappa, N)\big)$.

□

Proof. Case 2.

We modify the construction given in the previous proof and encode the input y in the entries of the coefficient matrix, with the goal of showing that $Q[\mathcal{U}_A] \in \Omega\big(\min(\kappa, N)\big)$. To this end, we define the vector $\mathbf{u} \in \mathbb{R}^{N+1}$ and a diagonal matrix $D \in \mathbb{R}^{(N+1)\times(N+1)}$

$$\begin{cases} u_i = 1 & \text{for } i \in [N] \\ u_{N+1} = \sqrt{N+M} \end{cases} \qquad \begin{cases} D_{i,i} = (-1)^{y_i} & \text{for } i \in [N] \\ D_{N+1,N+1} = 1 \end{cases} \tag{A.15}$$

and notice the vector $\mathbf{b}$ in Eq. (A.1) satisfies $\mathbf{b} = D\mathbf{u}$. We also define the coefficient matrix A'

$$A' := DAD \tag{A.16}$$

where A is given in Eq. (A.4). Note that D is unitary and self-inverse, hence A' is positive definite, $\kappa(A') = \kappa(A)$, and moreover $A'^{-1} = D^{-1}A^{-1}D^{-1} = DA^{-1}D$.

It is possible to implement exactly (i.e., ideally with zero error) a normalised matrix block of A' using at most 4 calls to $\mathcal{P}_y$. First, we consider the unitary Had$'$ that prepares the state $|\mathbf{1}'\rangle = \big|(1, 1, \ldots, 1, 0)^T\big\rangle$, that is Had $|0\rangle = |\mathbf{1}'\rangle$. Then, the matrix

$$\mathcal{U}_A := \begin{pmatrix} \text{Had} & 0 \\ 0 & \text{Had} \end{pmatrix} \begin{pmatrix} I - (1-\epsilon)\,|0\rangle\langle 0| & -\sqrt{1-(1-\epsilon)^2}\,|0\rangle\langle 0| \\ \sqrt{1-(1-\epsilon)^2}\,|0\rangle\langle 0| & I - (1-\epsilon)\,|0\rangle\langle 0| \end{pmatrix} \begin{pmatrix} \text{Had}^\dagger & 0 \\ 0 & \text{Had}^\dagger \end{pmatrix} \tag{A.17}$$

is a normalised matrix-block-encoding of $A = I - (1-\epsilon)K'$. Note that the matrix in the centre can be interpreted as the $|0\rangle\langle 0|$-controlled version of the Pauli-X rotation $e^{i\theta X}$ (with $\cos\theta = -(1-\epsilon)$) and is thus efficiently implementable. The operations in Eq. (A.2) correspond to a unitary quantum circuit that can be written as $D \oplus \mathcal{U}$, for some unitary $\mathcal{U}$, and finally we obtain that $\mathcal{U}_{A'} := (D \oplus \mathcal{U})\,\mathcal{U}_A\,(D \oplus \mathcal{U}^\dagger)$ is a matrix-block-encoding of A'.

We now consider the linear system $A'\mathbf{x} = \mathbf{u}$ and a quantum algorithm that prepares the corresponding solution state

$$\big|A'^{-1}\,\mathbf{u}\big\rangle = \big|DA^{-1}D\,\mathbf{u}\big\rangle = D\,\big|A^{-1}\,\mathbf{b}\big\rangle\,. \tag{A.18}$$

The state $D\,\big|A^{-1}\,\mathbf{b}\big\rangle$ can be transformed into $\big|A^{-1}\,\mathbf{b}\big\rangle$ using the steps given in Eq. (A.2), which only requires two extra accesses to $\mathcal{P}_y$. This state allows to solve PROMISEMAJORITY$_M$ with constant probability and thus the same considerations made in the preceding proof yield the result $Q[\mathcal{U}_A] \in \Omega\big(\min(\kappa, N)\big)$.

□

Proof. Case 3.

We start proving again a lower bound on the query complexity $Q[\mathcal{U}_\mathbf{b}]$, as in Case 1., but for a PD-QLS where the coefficient matrix A is sparse; then, we use the method used in the proof of Case 2. to convert it into a lower bound on $Q[\mathcal{P}_A]$.

Given $y \in \{0,1\}^N$ satisfying the PROMISEMAJORITY$_M$ condition, we introduce the known term vector $\mathbf{b} \in \mathbb{R}^{N+1}$

$$\begin{cases} b_i = (-1)^{y_i} & \text{for } i \in [N] \\ b_{N+1} = \sqrt{N}\,c_0 \end{cases} \tag{A.19}$$

where c_0 is a positive constant that we will fix later, and we have $|\mathbf{b}\rangle = \mathbf{b}\Big/\sqrt{N(1+c_0^2)}$.

Next, we define $B' \in \mathbb{R}^{(N+1)\times(N+1)}$ as

$$B' = B \oplus 0 \tag{A.20}$$

where $B \in \mathbb{R}^{N\times N}$ is a symmetric sparse matrix, having d non-zero entries in each row and column which are all equal to $\frac{1}{d}$, for some constant d. Since each row and column of B sums to one, B can be interpreted as the adjacency matrix of a Markov chain on a d-sparse graph. We require that the graph corresponding to B is not bipartite and that the spectral gap of B is large (i.e., the Markov chain is ergodic and rapidly mixing). These properties guarantee that B^t quickly converges to $K = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ for $t \to \infty$. Since B is symmetric, its spectrum is real and, because of the Perron-Froebenius theorem [52], the spectrum is contained in the interval $[-1, +1]$ and includes an eigenvalue $\lambda = 1$ with multiplicity one; the fact that B^t converges to $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ implies that -1 cannot be an eigenvalue of B. We then define the spectral gap $\delta(B)$ as the positive parameter $\delta(B) := \min_{\lambda\neq 1}\{1 - |\lambda|\}$, where the minimum is taken over the eigenvalues of B.

There are families of so-called *expander graphs* such that both the sparsity and the spectral gap are constant, see [53, Chapter 21]. We assume that B belongs to one of these expander families and thus, in particular, there is a (known) positive constant c_1 such that

$$\frac{1}{\delta(B)} \leq c_1 \tag{A.21}$$

for all sizes $N \in \mathbb{N}$. Moreover, $\mathbf{1}_N$ is the unique $+1$ eigenvector and hence, from the spectral decomposition of B, we can write

$$B = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T + \sum_{\lambda\neq 1}\lambda\,\mathbf{v}_\lambda\mathbf{v}_\lambda^\dagger = K + R\,. \tag{A.22}$$

Here $K = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$, $\mathbf{v}_\lambda$ are normalised eigenvectors of B, and thus $R \in \mathbb{R}^{N\times N}$ is a matrix of rank $N-1$ with $||R|| = 1 - \delta(B)$ and $KR = RK = 0$. Then we define:

$$A := I - (1-\epsilon)B' \tag{A.23}$$

for some $\epsilon > 0$ that we will define shortly. The matrix A can be used as a coefficient matrix in a PD-QLS solver since it is positive definite and with norm one. Moreover, A is $(d+1)$-sparse (where d is the sparsity of B, which is constant) and the condition number of A is exactly $\kappa(A) = 1/\epsilon$.

Next we have

$$A^{-1} = I + \sum_{t=1}^{\infty}\big((1-\epsilon)B'\big)^t \equiv I + \mathcal{B} \tag{A.24}$$

and then defining $\mathcal{K} := \sum_{t=1}^{\infty}\big((1-\epsilon)K'\big)^t = \frac{1-\epsilon}{\epsilon}K'$ we get

$$||\mathcal{B} - \mathcal{K}|| \leq \sum_{t=1}^{\infty}\Big|\Big|\big[(1-\epsilon)B\big]^t - \big[(1-\epsilon)K\big]^t\Big|\Big| \tag{A.25}$$

$$= \sum_{t=1}^{\infty}(1-\epsilon)^t\,\big|\big|(K+R)^t - K^t\big|\big| \tag{A.26}$$

$$= \sum_{t=1}^{\infty}(1-\epsilon)^t\,\big|\big|(K+R^t) - K\big|\big| \tag{A.27}$$

$$\leq \sum_{t=1}^{\infty}(1-\delta(B))^t \tag{A.28}$$

$$\leq \frac{1}{\delta(B)} \leq c_1 \tag{A.29}$$

where we have used $||R|| = 1 - \delta(B)$, $K^2 = K$ and $KR = RK = 0$.

Applying $A^{-1} = I + \mathcal{B}$ to $\mathbf{b}$ we thus obtain:

$$A^{-1}\mathbf{b} = \big[(I + \mathcal{B} - \mathcal{K}) + \mathcal{K}\big]\mathbf{b} \tag{A.30}$$

$$= (I + \mathcal{B} - \mathcal{K})\mathbf{b}' + \begin{cases} b_{N+1}\mathbf{e}_{N+1} + \frac{1-\epsilon}{\epsilon}\frac{M}{N}\,\mathbf{1}'_N & \text{if } y \text{ has a majority of 0} \\ b_{N+1}\mathbf{e}_{N+1} - \frac{1-\epsilon}{\epsilon}\frac{M}{N}\,\mathbf{1}'_N & \text{if } y \text{ has a majority of 1} \end{cases} \tag{A.31}$$

$$= (I + \mathcal{B} - \mathcal{K})\mathbf{b}' + \begin{cases} b_{N+1}\mathbf{e}_{N+1} + c_0\mathbf{1}'_N & \text{if } y \text{ has a majority of 0} \\ b_{N+1}\mathbf{e}_{N+1} - c_0\mathbf{1}'_N & \text{if } y \text{ has a majority of 1} \end{cases} \tag{A.32}$$

where $\mathbf{b}'$ is a vector equal to the first N entries of $\mathbf{b}$ (and $b'_{N+1}=0$), $\mathbf{1}'_N := (1,\dots,1,0)^T$ while $\mathbf{e}_{N+1}$ is the vector having a one in position $N+1$; moreover, we choose ϵ such that $\frac{1-\epsilon}{\epsilon}\frac{M}{N} = c_0$, where c_0 was introduced in the definition of $\mathbf{b}$ in Eq. (A.19). Then, fixing the constant c_0 as $c_0 = 100c_1$ and using the triangle inequality, we obtain the following upper bound

$$\sqrt{\mathcal{N}} = \big|\big|A^{-1}\mathbf{b}\big|\big| \leq \Big(\overbrace{c_0^2 N}^{|b_{N+1}|^2} + \overbrace{c_0^2 N}^{||c_0\mathbf{1}'_N||^2} \Big)^{1/2} + \overbrace{\sqrt{N}\,[1+c_1]}^{\geq ||(I+\mathcal{B}-\mathcal{K})\mathbf{b}'||} \tag{A.33}$$

$$\leq \left(\sqrt{c_0^2+c_0^2}+2\,c_1\right)\sqrt{N} \;=\; \Big(100\sqrt{2}+2\Big)\,c_1\,\sqrt{N} \tag{A.34}$$

$$\leq 144\,c_1\,\sqrt{N}, \tag{A.35}$$

where we have assumed $c_1 \geq 1$. We also have the lower bound

$$\sqrt{\mathcal{N}} = \big|\big|A^{-1}\mathbf{b}\big|\big| \geq \Big(\overbrace{c_0^2 N}^{|b_{N+1}|^2} + \overbrace{c_0^2 N}^{||c_0\mathbf{1}'_N||^2} \Big)^{1/2} - \overbrace{\sqrt{N}\,[1+c_1]}^{\geq ||(I+\mathcal{B}-\mathcal{K})\mathbf{b}'||} \tag{A.36}$$

$$\geq \Big(100\sqrt{2}-2\Big)\,c_1\,\sqrt{N} \tag{A.37}$$

$$\geq 139\,c_1\,\sqrt{N}. \tag{A.38}$$

Then we have, using $f = \textsc{PromiseMajority}_M(y)$,

$$\big|A^{-1}\mathbf{b}\big\rangle = \frac{A^{-1}\mathbf{b}}{||A^{-1}\mathbf{b}||} = \frac{c_0\sqrt{N}}{\sqrt{\mathcal{N}}}\,|N+1\rangle \;+\; (-1)^f\frac{c_0\sqrt{N}}{\sqrt{\mathcal{N}}}\sum_{i=1}^{N}\frac{1}{\sqrt{N}}\,|i\rangle + |\psi\rangle \tag{A.39}$$

$$= \frac{100}{144}\,|N+1\rangle \;+\; (-1)^f\frac{100}{144}\sum_{i=1}^{N}\frac{1}{\sqrt{N}}\,|i\rangle + |\psi'\rangle\,, \tag{A.40}$$

where $|\psi\rangle := (\mathcal{B}-\mathcal{K}+I)\mathbf{b}'/\sqrt{\mathcal{N}}$ is a sub-normalised perturbation vector with $\big|\big|\,|\psi\rangle\,\big|\big| \leq \frac{2}{139}$, while subtracting line (A.40) from line (A.39) one obtains

$$\Big|\Big|\;|\psi'\rangle - |\psi\rangle\;\Big|\Big| \leq \sqrt{2}\left(\frac{100}{139}-\frac{100}{144}\right) \leq 0.04 \tag{A.41}$$

and then we have:

$$\Big|\Big|\;|\psi'\rangle\;\Big|\Big| \leq 0.04 + \frac{2}{139} \leq 0.06\,. \tag{A.42}$$

The cases $f=0$ and $f=1$ in Eq. (A.40) can be distinguished with constant advantage using the swap test with the state $|\text{“}+\text{”}\rangle$ defined in Eq. (A.12), since we have:

$$\langle\text{“}+\text{”}|A^{-1}\mathbf{b}\rangle \;=\; \frac{1}{\sqrt{2}}\frac{100}{144} \;+\; (-1)^f\frac{1}{\sqrt{2}}\frac{100}{144} \;+\; \langle\text{“}+\text{”}|\,\psi'\,\rangle \tag{A.43}$$

$$\iff \begin{cases} \big|\,\langle\text{“}+\text{”}|A^{-1}\mathbf{b}\rangle\,\big| \;\geq\; 0.92 & \text{if } y \text{ has a majority of } 0 \\ \big|\,\langle\text{“}+\text{”}|A^{-1}\mathbf{b}\rangle\,\big| \;\leq\; 0.06 & \text{if } y \text{ has a majority of } 1\,. \end{cases} \tag{A.44}$$

Next, we proceed as in the proof of Case 2. and define an equivalent PD-QLS where the vector y is encoded in the entries of the coefficient matrix. We thus define the vector $\mathbf{u} \in \mathbb{R}^{N+1}$ and the diagonal matrix $D \in \mathbb{R}^{(N+1)\times(N+1)}$

$$\begin{cases} u_i = 1 & \text{for } i \in [N] \\ u_{N+1} = \sqrt{N}\,c_0 \end{cases} \qquad \begin{cases} D_{i,i} = (-1)^{y_i} & \text{for } i \in [N] \\ D_{N+1,N+1} = 1 \end{cases} \tag{A.45}$$

and thus the identity $\mathbf{b} = D\mathbf{u}$ holds. We then introduce A', given by

$$A' := DAD \tag{A.46}$$

where A is as in Eq. (A.23). The matrix D is unitary and self-inverse, hence $\kappa(A') = \kappa(A)$, and $A'^{-1} = D^{-1}A^{-1}D^{-1} = DA^{-1}D$.

Notice that the position of the non-zero entries of A' are the same as in A and thus independent from y, while the sign of a non-zero entry $A'_{i,j} = (-1)^{y_i+y_j}$ can be obtained querying $\mathcal{P}_y$ once with input $|i\rangle$ and once with input $|j\rangle$. These queries can be performed in quantum superposition and thus two accesses to $\mathcal{P}_y$ are sufficient to implement a quantum sparse-matrix-access $\mathcal{P}'_A$. Therefore it is possible to prepare, with the same $\mathcal{P}_y$-complexity as discussed previously, the state

$$\left|A'^{-1}\,\mathbf{u}\right\rangle = \left|DA^{-1}D\,\mathbf{u}\right\rangle = D\left|A^{-1}\,\mathbf{b}\right\rangle . \tag{A.47}$$

Finally, we can obtain $\left|A^{-1}\,\mathbf{b}\right\rangle$ using two extra accesses to $\mathcal{P}_y$ by applying the transformations given in Eq. (A.2). This proves that a quantum algorithm that solves the PD-QLS having access to $\mathcal{P}'_A$ necessarily has a query complexity $Q[\mathcal{P}_{A'}] \in \Omega\big(\min(\kappa, N)\big)$.

□

B Scaling of the normalisation factor of the matrix-block-encoding

In this Appendix, we consider the polynomial

$$P_{2\ell-1,\kappa}(x) := \frac{1}{1-x}\left[1-\hat{\mathcal{T}}_{\ell,\kappa}(x)\right]^2 \tag{B.1}$$

as was defined in Eq. (26) and where we have

$$\hat{\mathcal{T}}_{\ell,\kappa}(x) := \frac{\mathcal{T}_\ell\left(\frac{x+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)}{\mathcal{T}_\ell\left(\frac{1+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)} . \tag{B.2}$$

We prove that the normalisation factor $K := 2\max_{x\in[-1,+1]} P_{2\ell-1,\kappa}(x)$ satisfies $K \in \Theta(\kappa)$, provided that $\ell \geq c\sqrt{\kappa}$ for some constant c that we will determine later.

Notice that by construction $P_{2\ell-1,\kappa}(1) = 0$ and the polynomial is positive for $x \in [-1,+1)$. To study the properties of the local maxima of $P_{2\ell-1,\kappa}(x)$ in the interval $[-1,+1]$ we compute the derivative of $P(x)$ using the property $\frac{\partial}{\partial x}\mathcal{T}_\ell(x) = \ell\,\mathcal{U}_{\ell-1}(x)$, where $\mathcal{U}_\ell(x) \in \mathbb{R}_\ell[x]$ is a Chebyshev polynomial of the second kind. We have:

$$\frac{\partial P_{2\ell-1,\kappa}(x)}{\partial x} = \frac{\left[1-\hat{\mathcal{T}}_{\ell,\kappa}(x)\right]^2}{(1-x)^2} - 2\ell\,\frac{1-\hat{\mathcal{T}}_{\ell,\kappa}(x)}{1-x}\,\frac{\mathcal{U}_{\ell-1}\left(\frac{x+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)}{\left(1-\frac{1}{2\kappa}\right)\mathcal{T}_\ell\left(\frac{1+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)} . \tag{B.3}$$

We set the derivative equal to 0 and simplify the expression assuming $1-x \neq 0$ and $1-\hat{\mathcal{T}}_{\ell,\kappa}(x) \neq 0$:

$$\left(1-\frac{1}{2\kappa}\right)\left[\mathcal{T}_\ell\left(\frac{1+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right) - \mathcal{T}_\ell\left(\frac{x+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right)\right] - 2\ell\,(1-x)\,\mathcal{U}_{\ell-1}\left(\frac{x+\frac{1}{2\kappa}}{1-\frac{1}{2\kappa}}\right) = 0 \tag{B.4}$$

Then we use the change of variables $y(x)$ and $\delta(\kappa)$ given in Eqs. (22) and their inverses $\kappa(\delta) = \frac{1}{\delta}+\frac{1}{2}$, $x(y) = \frac{y-\delta/2}{1+\delta/2}$ to rewrite the previous equation as

$$\left(\frac{1}{1+\delta/2}\right)\left[\mathcal{T}_\ell(1+\delta) - \mathcal{T}_\ell(y)\right] - 2\ell\left(\frac{1+\delta-y}{1+\delta/2}\right)\mathcal{U}_{\ell-1}(y) = 0 \tag{B.5}$$

which is equivalent to:

$$\boxed{\mathcal{T}_\ell(1+\delta) - \mathcal{T}_\ell(y) \;=\; 2\ell\,(1+\delta-y)\,\mathcal{U}_{\ell-1}(y)} \tag{B.6}$$

for $x \in [-1,+1]$ or, equivalently, $y \in [-1, 1+\delta]$.

The polynomial $P_{2\ell-1,\kappa}(x)$ is by construction non-negative on the domain $x \in [-1,+1]$ and its derivative in $x = 1-\frac{1}{\kappa}$ (corresponding to $y = 1$) is positive, provided that $\ell \in \Omega(1/\sqrt{\delta})$. In

fact, the derivative of $P_{2\ell-1,\kappa}(x)$ is positive if and only if the left hand side of Eq. (B.6) is larger than the right hand side; using $\mathcal{T}_\ell(1+\delta) \geq \frac{1}{2}e^{\ell\sqrt{\delta}}$ for $0 \leq \delta \leq 3-2\sqrt{2}$, $\mathcal{T}_\ell(1)=1$, $\mathcal{U}_{\ell-1}(1)=\ell$ we obtain from (B.6) the inequality $\frac{1}{2}e^{\ell\sqrt{\delta}}-1 \overset{!}{>} 2\,\ell^2\delta$, which is satisfied for $\ell \geq 4.36/\sqrt{\delta}$. Since we have, moreover, $P_{2\ell-1,\kappa}(1)=0$, the function $P_{2\ell-1,\kappa}(x)$ does not have any local maximum in $\left[-1, 1-\frac{1}{\kappa},\right]$ and must have one or more local maxima $x_* \in \left(1-\frac{1}{\kappa}, +1\right]$; equivalently, Eq. (B.6) must have at least one solution $y_* \in (1, 1+\delta]$.

Any local maximum $y_* = y(x^*)$ satisfies:

$$\mathcal{T}_\ell(y_*) = \mathcal{T}_\ell(1+\delta) - 2\ell\,(1+\delta-y_*)\,\mathcal{U}_{\ell-1}(y_*) \tag{B.7}$$

and substituting $\mathcal{T}_\ell(y_*)$ in the definition (B.1) gives

$$P_{2\ell-1,\kappa}(x_*) = \frac{1+\delta/2}{1+\delta-y_*}\left[1-\frac{\mathcal{T}_\ell(y_*)}{\mathcal{T}_\ell(1+\delta)}\right]^2 \tag{B.8}$$

$$= 4\ell^2(1+\delta/2)(1+\delta-y_*)\,\frac{\mathcal{U}_{\ell-1}(y_*)^2}{\mathcal{T}_\ell(1+\delta)^2}\,. \tag{B.9}$$

From Eq. (B.6) we directly have

$$\mathcal{U}_{\ell-1}(y_*) \leq \frac{\mathcal{T}_\ell(1+\delta)}{2\ell(1+\delta-y_*)} \tag{B.10}$$

and inserting this inequality in Eq. (B.9) we have that any local maximum x_* satisfies

$$P_{2\ell-1,\kappa}(x_*) \leq \frac{1+\delta/2}{1+\delta-y_*} \tag{B.11}$$

$$\leq \frac{3/2}{1+\delta-y_*}\,. \tag{B.12}$$

In summary, it will be sufficient to show $1+\delta-y_* \in \Omega(\delta)$ to prove that the normalisation constant satisfies $K = 2\,\max_{\{x_*\}}|P_{2\ell-1,\kappa}(x_*)| \in \mathcal{O}(\kappa)$, where the maximisation is over the set of (potentially multiple) local maxima $x_* \in \left[1-\frac{1}{\kappa}, +1\right]$.

To this end, we rewrite Eq. (B.6) as:

$$\frac{\mathcal{T}_\ell(1+\delta)-\mathcal{T}_\ell(y_*)}{(1+\delta)-y_*} \;=\; 2\,\frac{\partial\mathcal{T}_\ell}{\partial y}(y_*)\,. \tag{B.13}$$

Since both the first and the second derivative of $\mathcal{T}_\ell(y)$ are positive for $y \geq 1$ (i.e., the derivative of $\mathcal{T}_\ell(y)$ is monotonically increasing), this equation can be satisfied only if[14]

$$2\frac{\partial\mathcal{T}_\ell}{\partial y}(y_*) \overset{!}{\leq} \frac{\partial\mathcal{T}_\ell}{\partial y}(1+\delta) \tag{B.14}$$

or equivalently, defining $1+\delta_* := y_*$

$$2\,\mathcal{U}_{\ell-1}(1+\delta_*) \overset{!}{\leq} \mathcal{U}_{\ell-1}(1+\delta)\,. \tag{B.15}$$

A Chebyshev polynomial of the second kind can be written as

$$\mathcal{U}_{\ell-1}(y) = \frac{(y+\sqrt{y^2-1})^\ell - (y+\sqrt{y^2-1})^{-\ell}}{2\sqrt{y^2-1}} \tag{B.16}$$

hence we have

$$\mathcal{U}_{\ell-1}(1+\delta_*) \leq \frac{(1+\delta_*+\sqrt{2\delta_*+\delta_*^2})^\ell}{2\sqrt{2\delta_*+\delta_*^2}} \tag{B.17}$$

$$\leq \frac{(1+1.1\sqrt{2\,\delta_*})^\ell}{2\sqrt{2\delta_*}} \tag{B.18}$$

[14] In this appendix we employ the notation $\overset{!}{\leq}$ to mark inequalities that we still need to prove and with $\leq$ an inequality that has already been proven.

for sufficiently small δ_* (the constant 1.1 is somewhat arbitrary), while we have

$$\mathcal{U}_{\ell-1}(1+\delta) = \frac{(1+\delta+\sqrt{2\delta+\delta^2})^\ell\left[1-(1+\delta+\sqrt{2\delta+\delta^2})^{-2\ell}\right]}{2\sqrt{2\delta+\delta^2}} \tag{B.19}$$

$$\geq \frac{(1+\sqrt{2\delta})^\ell \times 0.8}{2.4\sqrt{2\delta}} \tag{B.20}$$

$$= \frac{2}{3}\frac{(1+\sqrt{2\delta})^\ell}{2\sqrt{2\delta}} \tag{B.21}$$

$$\geq \frac{2}{3}\frac{(1+\sqrt{2\delta})^\ell}{2\sqrt{3\delta_*}} \tag{B.22}$$

which holds for $\ell \geq 1/\sqrt{2\delta}$ and in the last step we have assumed $\delta_* \geq \frac{2}{3}\delta$ (notice that in the opposite case $\delta_* < \frac{2}{3}\delta$ we would already have $1+\delta-y_* = \delta-\delta_* > \frac{1}{3}\delta$). Thus, the inequality in (B.15) is implied by

$$2\,\frac{(1+1.1\sqrt{2\,\delta_*})^\ell}{2\sqrt{2\delta_*}} \overset{!}{\leq} \frac{2}{3}\frac{(1+\sqrt{2\delta})^\ell}{2\sqrt{3\delta_*}} \tag{B.23}$$

$$\iff \quad (1+1.1\sqrt{2\,\delta_*})^\ell \overset{!}{\leq} \frac{\sqrt{2}}{3\sqrt{3}}\,(1+\sqrt{2\delta})^\ell \tag{B.24}$$

$$\iff \quad 1+1.1\sqrt{2\,\delta_*} \overset{!}{\leq} e^{-\frac{1}{\ell}\log(3\sqrt{3/2})}\,(1+\sqrt{2\delta})\,. \tag{B.25}$$

From the inequality $e^{-x} \geq 1-x$ for $x \geq 0$ we see that the inequality above is implied by

$$1.1\sqrt{2\,\delta_*} \overset{!}{\leq} \left(1-1.31/\ell\right)(1+\sqrt{2\delta})-1 \tag{B.26}$$

$$= \sqrt{2\delta} - \frac{1}{\ell}1.31\,(1+\sqrt{2\delta}) \tag{B.27}$$

with $1.31 \geq \log(3\sqrt{3/2})$. We now make the assumption that $\frac{1}{\ell}1.31\,(1+\sqrt{2\delta}) \leq \frac{1}{10}\sqrt{2\delta}$, which is implied by $\ell \geq 13.1 + 9.27/\sqrt{\delta}$ and is compatible with the requirement $\ell \in \Omega(\delta^{-1/2})$. Thus it is sufficient to impose:

$$1.1\sqrt{2\,\delta_*} \overset{!}{\leq} \frac{9}{10}\sqrt{2\delta} \tag{B.28}$$

$$\iff \quad \delta_* \overset{!}{\leq} \left(\frac{9}{11}\right)^2 \delta\,. \tag{B.29}$$

We also remark that $(9/11)^2 \geq 2/3$, so that this last inequality is compatible with the assumption $\delta_* \geq \frac{2}{3}\delta$ we made earlier.

Plugging the bound in Eq. (B.29) into (B.11), together with the definition of $\delta(\kappa)$, results in the explicit bound $K \leq 6.05\,\kappa$, i.e. $K \in \mathcal{O}(\kappa)$, provided that $\ell \geq 13.1 + 9.27\sqrt{\kappa - 1/2}$, i.e. $\ell \in \Omega(\sqrt{\kappa})$.

C VTAA optimization improving the runtime values of the first algorithm

In this Appendix we use VTAA to speed-up the asymptotic runtime of the algorithm based on polynomial approximations of $1/(1-x)$. Preliminarily, we introduce a Lemma showing that this VTAA algorithm, as summarised in Proposition 13, does provide an asymptotic query complexity improvement (compared to general QLS solvers) for most values of $\Gamma_{A,\mathbf{b}}$, see the right plot in Figure 3.

Lemma 20. *Defining* $\Gamma_{A,\boldsymbol{b}} := \sqrt{\kappa}\,\frac{||A^{-1/2}|\boldsymbol{b}\rangle||}{||A^{-1}|\boldsymbol{b}\rangle||}$ *we have* $\Gamma_{A,\boldsymbol{b}} \in [1,\sqrt{\kappa}\,]$*, under the usual assumption that the spectrum of* A *is contained in* $[1/\kappa, 1]$.

Proof. We expand $|\mathbf{b}\rangle$ in a basis of eigenvectors of A, that is $|\mathbf{b}\rangle = \sum_\lambda \beta_\lambda |\lambda\rangle$ with $A|\lambda\rangle = \lambda|\lambda\rangle$, $\langle\lambda|\lambda'\rangle = \delta_{\lambda,\lambda'}$, and $\sum_\lambda |\beta_\lambda|^2 = 1$. Then, we can write

$$\frac{\left|\left|A^{-1/2}|\mathbf{b}\rangle\right|\right|^2}{||A^{-1}|\mathbf{b}\rangle||^2} = \frac{\sum_\lambda |\beta_\lambda|^2 f_\lambda}{\sum_\lambda |\beta_\lambda|^2 g_\lambda} \in \left[\min_\lambda\{f_\lambda/g_\lambda\}, \max_\lambda\{f_\lambda/g_\lambda\}\right] \subseteq \left[\kappa^{-1}, 1\right] \tag{C.1}$$

for $f_\lambda = \lambda^{-1}$, $g_\lambda = \lambda^{-2}$ and thus $f_\lambda/g_\lambda = \lambda \in [\kappa^{-1}, 1]$. We take the square root of both extrema and multiply by $\sqrt{\kappa}$ to conclude.

□

C.1 Variable-time amplitude amplification review

In this Section we review the general VTAA method, mainly following Ref. [19, Section 3].

Definition 21 (Variable-stopping-time quantum algorithm)**.** *A variable-stopping-time quantum algorithm $\mathcal{A} = \mathcal{A}_m \cdot \ldots \cdot \mathcal{A}_1 \cdot \mathcal{A}_0$ is given by the application of $m+1$ sub-algorithms $\mathcal{A}_j$ in sequence, acting on the Hilbert space $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_F \otimes \mathcal{H}_S$ where S is a "system register" of arbitrary size, F is a "flag qubit" that heralds success and $\mathcal{H}_C = \bigotimes_{j=0}^m \mathcal{H}_{C_j}$ is a "clock register" containing $m+1$ qubits $C_0, C_1, \ldots, C_m$. $\mathcal{H}$ is initially prepared in the all-zero state $|0\rangle_{\text{all}} = |0,0,0\rangle_{C,F,S}$. Each $\mathcal{A}_j$ acts on $\mathcal{H}_{C_j} \otimes \mathcal{H}_F \otimes \mathcal{H}_S$ and $|1\rangle_{C_j}$ indicates that the algorithm stops after the application of $\mathcal{A}_j$; i.e., each $\mathcal{A}_j$ is a controlled algorithm that acts if and only if the previous j qubits in the clock register are in the state $|0\rangle^{\otimes j} \in \bigotimes_{i=0}^{j-1} \mathcal{H}_{C_i}$. We assume that all branches of the computation end by step m. The successful branches of the algorithm are those where the flag is in the state $|1\rangle_F$ and thus we define*

$$p_{\text{succ}} := \left|\left|\,\Pi_\checkmark \mathcal{A}\,|0\rangle_{\text{all}}\,\right|\right|^2 \qquad \text{and} \qquad |\psi_{\text{succ}}\rangle := \frac{\Pi_\checkmark \mathcal{A}\,|0\rangle_{\text{all}}}{\left|\left|\,\Pi_\checkmark \mathcal{A}\,|0\rangle_{\text{all}}\,\right|\right|} \tag{C.2}$$

where $\Pi_\checkmark := I_C \otimes |1\rangle\langle 1|_F \otimes I_S$.

By construction, a variable-stopping-time quantum algorithm produces a quantum state of the form $\mathcal{A}|0\rangle_{\text{all}} = \sum_{j=0}^m \alpha_j |1_j\rangle_C |\Psi_j\rangle_{F,S}$, where

$$|1_j\rangle_C := |0\rangle^{\otimes m-j} |1\rangle |0\rangle^{\otimes j} \tag{C.3}$$

is a state having the qubit C_j in $|1\rangle$ and the other clock qubits in $|0\rangle$, while $|\Psi_j\rangle_{F,S}$ are normalised quantum states in $\mathcal{H}_F \otimes \mathcal{H}_S$ with $\sum_{j=0}^m |\alpha_j|^2 = 1$.

Definition 22 (Stopping times)**.** *We introduce a sequence of* stopping times $t_{\min} \equiv t_0 < t_1 < t_2 < \ldots < t_m \equiv t_{\max}$*, where each $t_j \in \mathbb{N}$ is the complexity of the sub-algorithm $\mathcal{A}_{\leq j} := \mathcal{A}_j \cdot \ldots \cdot \mathcal{A}_0$. The probability p_j of stopping at time t_j (i.e., after the execution of algorithm $\mathcal{A}_{\leq j}$) and the ℓ_2-average* stopping time *are defined as*

$$p_j := \left|\left|\,\Pi_{C_j} \mathcal{A}_{\leq j}\,|0\rangle_{\text{all}}\,\right|\right|^2 \qquad t_{\text{avg}} := \sqrt{\sum_{j=0}^m p_j\, t_j^2} \tag{C.4}$$

with $\Pi_{C_j} := |1_j\rangle\langle 1_j|_C \otimes I_F \otimes I_S$. We define $\Pi_{\text{stop}\leq j} := \sum_{i=0}^j \Pi_{C_i}$ and

$$\Pi_{\text{bad}}^j := \Pi_{\text{stop}\leq j} \cdot \left(I_C \otimes |0\rangle\langle 0|_F \otimes I_S\right) \tag{C.5}$$

$$\Pi_{\text{mg}}^j := I - \Pi_{\text{bad}}^j \tag{C.6}$$

which project onto "bad" and "maybe good" subspaces after the application of $\mathcal{A}_{\leq j}$. The "maybe good" subspace at step $j+1$ is contained in the "maybe good" subspace at step j and, by construction, $\Pi_{\text{mg}}^m \mathcal{A}|0\rangle_{\text{all}} = \Pi_\checkmark \mathcal{A}|0\rangle_{\text{all}}$.

In the definition above the t_j can represent any complexity measure (e.g., query complexity, gate complexity, circuit depth). In this Appendix we only compute the query complexity Q of the sub-algorithm $\mathcal{A}_{\leq j}$, but we remark that all the algorithms we consider here are gate-efficient, i.e., the gate complexity is in $\mathcal{O}\big(Q\,\text{poly}(\log Q, \log N)\big)$.

Definition 23 (Variable-Time Amplitude Amplification)**.** *Given a variable-stopping-time algorithm $\mathcal{A} = \mathcal{A}_m \cdot \ldots \cdot \mathcal{A}_0$ as in Definition 21 and given a sequence of $m+1$ non-negative integers $(k_0, k_1, \ldots, k_m)$, we recursively define a variable-time amplification $\mathcal{A}' = \mathcal{A}'_m \cdot \ldots \cdot \mathcal{A}'_0$ as follows. Setting $\mathcal{A}_{-1} := I$, each $\mathcal{A}'_j$ implements a standard k_j-step amplitude amplification that uses $\mathcal{A}_j\mathcal{A}'_{j-1}$ and its inverse a total of $2k_j+1$ times, where the input state is $|\psi^j_{\text{in}}\rangle = \mathcal{A}_j\mathcal{A}'_{j-1}|0\rangle_{\text{all}}$ and the target state is $\Pi^j_{\text{mg}}|\psi^j_{\text{in}}\rangle$. That is, $\mathcal{A}'_j$ begins preparing the input state*

$$|\psi^j_{\text{in}}\rangle = \mathcal{A}_j\mathcal{A}'_{j-1}|0\rangle_{\text{all}} = \sin(\theta_j)\,|\psi^j_{\text{mg}}\rangle + \cos(\theta_j)\,|\psi^j_{\text{bad}}\rangle \tag{C.7}$$

$$\text{with } \begin{cases} |\psi^j_{\text{mg}}\rangle \propto \Pi^j_{\text{mg}}|\psi^j_{\text{in}}\rangle \\ |\psi^j_{\text{bad}}\rangle \propto \Pi^j_{\text{bad}}|\psi^j_{\text{in}}\rangle \end{cases} \qquad \theta_j := \arcsin\big(\big\|\Pi^j_{\text{mg}}\mathcal{A}_j\mathcal{A}'_{j-1}|0\rangle_{\text{all}}\big\|\big) \in [0, \pi/2] \tag{C.8}$$

and then uses k_j accesses to the reflections $R^j_{\text{out}} := I - 2\,\Pi^j_{\text{mg}}$ and $R^j_{\text{in}} := 2\,|\psi^j_{\text{in}}\rangle\langle\psi^j_{\text{in}}| - I$, where R^j_{in} is implemented using $\mathcal{A}_j\mathcal{A}'_{j-1}$ and $(\mathcal{A}_j\mathcal{A}'_{j-1})^\dagger$ once, to obtain the output state

$$|\psi^j_{\text{out}}\rangle = \mathcal{A}'_j|0\rangle_{\text{all}} = \sin\big[(2k_j+1)\theta_j\big]\,|\psi^j_{\text{mg}}\rangle + \cos\big[(2k_j+1)\theta_j\big]\,|\psi^j_{\text{bad}}\rangle\,. \tag{C.9}$$

Note that if we set $k_0 = \ldots = k_m = 0$ we have $\mathcal{A}' \equiv \mathcal{A}$. Also, by the recursive structure of VTAA the first sub-algorithm $\mathcal{A}_1$ is used a total of $\prod_{j=0}^m(2k_j+1)$ times in $\mathcal{A}'$, which grows exponentially in m if $k_j \geq 1$. Nonetheless, VTAA can provide a speed-up when the amplification parameters $(k_0, k_1, \ldots, k_m)$ are chosen appropriately. More precisely, the following result can be derived from Ref. [19, Lemma 22].

Proposition 24 (Result of VTAA)**.** *Using the notation of the previous definitions, let $\mathcal{A}'$ be a variable-time amplification such that each $\mathcal{A}'_j$ uses k_j steps of amplitude amplification, where*

$$\frac{\pi}{8\,\theta_j} - \frac{1}{2} \leq k_j \leq \frac{\pi}{4\,\theta_j} - \frac{1}{2} \tag{C.10}$$

Then, with the definitions in Eq. (C.2), $\mathcal{A}'$ outputs the state

$$\mathcal{A}'|0\rangle_{\text{all}} = \sqrt{p'_{\text{succ}}}\,|\psi_{\text{succ}}\rangle_{C,F,S} + \sqrt{1-p'_{\text{succ}}}\,|\psi^\perp\rangle_{C,F,S} \tag{C.11}$$

where success is heralded by $|1\rangle_F$, the success probability satisfies $p'_{\text{succ}} \in \Theta(1)$, and the global query complexity is

$$Q' \in \mathcal{O}\left(t_{\max}\sqrt{m} + \frac{t_{\text{avg}}}{\sqrt{p_{\text{succ}}}}\sqrt{m\,\log(t_{\max}/t_{\min})}\right). \tag{C.12}$$

We can compare Eq. (C.12) with standard amplitude amplification, which has a query complexity $Q \in \mathcal{O}(t_{\max}/\sqrt{p_{\text{succ}}})$. In our algorithm, m will scale logarithmically in $t_{\max}$, so the total runtime is $\widetilde{\mathcal{O}}(t_{\max} + t_{\text{avg}}/\sqrt{p_{\text{succ}}})$. Hence, VTAA can provide a speed-up when the average runtime is much shorter than the maximum runtime.

We remark that a sequence of good values $(k_0, \ldots, k_m)$ such that conditions in Eq. (C.10) are satisfied can be obtained efficiently by means of an iterative classical-quantum pre-processing algorithm. Specifically, suppose that some values $(k_0, \ldots, k_j)$ satisfying (C.10) have been already found; then, one can compile the corresponding VTAA algorithm $\mathcal{A}'_j$ and use phase estimation on the state $|\psi^{j+1}_{\text{in}}\rangle = \mathcal{A}_{j+1}\mathcal{A}'_j|0\rangle_{\text{all}}$ to obtain an estimate of $\sin(\theta_{j+1})$ up to constant multiplicative precision; this allows to find a value k_{j+1} which satisfies (C.10) with probability $1-p_{\text{fail}}$ with a cost $\mathcal{O}\big(\frac{1}{\theta_{j+1}}\log(1/p_{\text{fail}})\big)$ [30], which is asymptotically equal to the query complexity of $\mathcal{A}'_{j+1}$, apart from a multiplicative $\log(1/p_{\text{fail}})$ overhead. Once k_{j+1} has been precomputed one can directly compile $\mathcal{A}'_{j+1}$, without the need of performing phase estimation "online". The total failure probability is $p^{\text{tot}}_{\text{fail}} \leq m\,p_{\text{fail}}$ and the query cost of the hybrid classical-quantum pre-processing algorithm has only a $\mathcal{O}\big(\log(1/p_{\text{fail}})\big)$ multiplicative overhead compared to the expression in Eq. (C.12).

C.2 Efficient implementation of windowing polynomials

In this Section we introduce the "windowing functions" that we use to replace the Gapped Phase Estimation (GPE) subroutine, introduced in Ref. [5]. Using GPE gives an additive $\mathcal{O}(\kappa)$ runtime overhead, which is too costly for our purposes.

Lemma 25 (Efficient windowing polynomials)**.** *Given $\epsilon, \delta \in (0, 1/2]$, there exists an even polynomial $W_{\epsilon,\delta}(x) = W_{\epsilon,\delta}(-x)$ of degree $\ell \in \mathcal{O}\Big(\frac{1}{\sqrt{\delta}} \operatorname{polylog}(\delta^{-1}, \epsilon^{-1})\Big)$, where $\operatorname{polylog}(\delta^{-1}, \epsilon^{-1}) \equiv \log^{1/4}(\epsilon^{-1})\big[\log(\epsilon^{-1}) + \log(\delta^{-1})\big]$, satisfying the inequalities*

$$W_{\epsilon,\delta}(x) \in \begin{cases} [1-\epsilon, 1] & \text{if } x \in [0, 1-2\delta] \\ [-1, +1] & \text{if } x \in (1-2\delta, 1-\delta) \\ [-\epsilon, +\epsilon] & \text{if } x \in [1-\delta, 1]\,. \end{cases} \tag{C.13}$$

The windowing function $W_{\epsilon,\delta}(x)$ can be computed efficiently with classical algorithms.

A family of polynomials satisfying the inequalities in (C.13) is already given in Ref. [25, Lemma 29], but they have a degree $\ell \in \widetilde{\mathcal{O}}(\delta^{-1})$. We seek to achieve the same result with a quadratically smaller degree, $\ell \in \widetilde{\mathcal{O}}(\delta^{-1/2})$. Also, it is crucial that the polynomial approximation has even parity: using QSP one can implement matrix polynomials when the polynomial P satisfies $|P(x)| \leq 1$ for $|x| \leq 1$ and P has definite parity, while for a polynomial P' without definite parity the more restrictive constraint $|P'(x)| \leq 1/2$ has to be satisfied (see Theorem 8).

Proof. The proof proceeds in two steps: first, we find an analytic function $f(x)$ that satisfies the inequalities in (C.13) within error $\epsilon/2$, and then show that the Chebyschev expansion converges very quickly to it, so that choosing a degree $\ell \in \widetilde{\mathcal{O}}\big(\delta^{-1/2}\big)$ is sufficient to be within $\epsilon/2$-distance from $f(x)$ for all x.

First part: We introduce the normal distribution cumulative function

$$\Phi(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt \qquad x \in \mathbb{R} \tag{C.14}$$

normalised so that $0 \leq \Phi(x) \leq 1$. We then define

$$\mathcal{W}_{\sigma,\delta}(x) := \Phi\left(\frac{x + 1 - 1.5\,\delta}{\sigma}\right) \Phi\left(\frac{-x + 1 - 1.5\,\delta}{\sigma}\right) \tag{C.15}$$

where $\sigma > 0$ has to be chosen so that $\mathcal{W}_{\sigma,\delta}(1-2\delta) \geq 1 - \epsilon/2$ and $\mathcal{W}_{\sigma,\delta}(1-\delta) \leq \epsilon/2$. Using $\Phi(-x) = 1 - \Phi(x)$ and the monotonicity of Φ, both inequalities are implied by $\Phi\big(-\frac{0.5\,\delta}{\sigma}\big) \leq \epsilon/4$. Using the bound $\Phi(-x) \leq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-x} \frac{-t}{x} e^{-t^2/2}\, dt = \frac{e^{-x^2/2}}{\sqrt{2\pi}x} \leq \frac{e^{-x^2/2}}{\sqrt{2\pi}}$ for $x \geq 1$, it is sufficient to choose $\sigma \in \Theta\big(\delta/\sqrt{\log(\epsilon^{-1})}\big)$ to obtain the desired result.

Second part: We consider the Chebyschev series associated to $\mathcal{W}_{\sigma,\delta}(x)$. Note that the function $\mathcal{W}_{\sigma,\delta}(x)$ has even parity, hence also the associated Chebyschev series has even parity [54]. Truncating the Chebyschev series at degree ℓ, we get a sequence of polynomials $[S_\ell \mathcal{W}_{\sigma,\delta}](x)$ converging uniformly to $\mathcal{W}_{\sigma,\delta}(x)$ for $\ell \to \infty$. More precisely, we have the following result [54, Theorem 5.16].

Lemma 26. *Suppose $f(x)$ can be extended to an analytic function on the ellipse $E_r \subseteq \mathbb{C}$*

$$E_r = \Big\{\, a\cos\theta + i\, b \sin\theta \;\Big|\; a = \tfrac{1}{2}(r + r^{-1}),\ b = \tfrac{1}{2}(r - r^{-1}),\ \theta \in [0, 2\pi) \,\Big\} \tag{C.16}$$

for some $r > 1$; then, the ℓ-th degree truncation of the Chebyschev series is a polynomial $[S_\ell f](x)$ that satisfies for all $x \in [-1, +1]$

$$\Big|f(x) - [S_\ell f](x)\Big| \leq \frac{M}{r^\ell (r-1)} \qquad \text{with } M = \sup\big\{|f(z)| : z \in E_r\big\}. \tag{C.17}$$

We apply this theorem to $f(x) = \mathcal{W}_{\sigma,\delta}(x)$, which is analytic in the whole complex plane. The main technical hurdle is to upper bound M. We choose $r = \frac{1+\sqrt{\sigma}}{1-\sqrt{\sigma}} \geq 1 + 2\sqrt{\sigma}$, so that we have $a = \frac{1+\sigma}{1-\sigma}$ and $b = \frac{2\sqrt{\sigma}}{1-\sigma}$, where a and b are the semi-axis of the ellipse E_r along the real and imaginary axis, respectively. We have, for $x, y \in \mathbb{R}$:

$$|\Phi(x+iy)| = \tfrac{1}{\sqrt{2\pi}} \Big| \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt + \int_{x}^{x+iy} e^{-\frac{t^2}{2}}\, dt \Big| \tag{C.18}$$

$$\leq 1 + \tfrac{1}{\sqrt{2\pi}} \Big| \int_{0}^{iy} e^{-\frac{(x+i\tau)^2}{2}}\, d\tau \Big| \tag{C.19}$$

$$\leq 1 + \tfrac{1}{\sqrt{2\pi}}\, e^{-x^2/2} \int_{0}^{|y|} e^{\frac{\tau^2}{2}}\, d\tau \tag{C.20}$$

$$\leq 1 + \tfrac{1}{\sqrt{2\pi}}\, e^{-x^2/2}\, e^{y^2/2}\, |y| \tag{C.21}$$

$$\leq 1 + \tfrac{1}{\sqrt{2\pi}} e^{(-x^2+y^2)/2} \tag{C.22}$$

where the last inequality holds for $|y| \leq 1$. We now upper bound

$$M_{\text{half}} := \sup \Big\{ \Big| \Phi\Big(\tfrac{z+1-1.5\,\delta)}{\sigma} \Big) \Big| : z \in E_r \Big\} \tag{C.23}$$

so that we will have $M = \sup_{E_r} |\mathcal{W}_{\sigma,\delta}(z)| \leq M_{\text{half}}^2$. From (C.22) we only need to maximize the expression $1 + \frac{1}{\sqrt{2\pi}} e^{(-x^2+y^2)/2}$ for

$$x = \frac{a}{\sigma}\cos\theta + \frac{1-1.5\delta}{\sigma} \qquad y = \frac{b}{\sigma}\sin\theta\,. \tag{C.24}$$

Equivalently, we can maximize $-x^2 + y^2$. Substituting $\cos\theta = u$ and $\sin^2\theta = 1 - u^2$ we get

$$-x^2 + y^2 = \frac{1}{\sigma^2}\big[-(au + 1 - 1.5\delta)^2 + b^2(1-u^2) \big] \tag{C.25}$$

$$\leq \frac{b^2}{2\sigma^2} \left[1 - \frac{(1-1.5\delta)^2}{a^2+b^2} \right]. \tag{C.26}$$

We now have $b \in \mathcal{O}(\sqrt{\sigma})$, $\frac{1}{a^2+b^2} = \frac{1-2\sigma+\sigma^2}{1+6\sigma+\sigma^2} = 1 - \mathcal{O}(\sigma)$ and thus

$$\sup\{-x^2+y^2\} \;\in \mathcal{O}\Big(\frac{\sigma}{\sigma^2}\Big)\Big(1 - [1-\mathcal{O}(\delta)][1-\mathcal{O}(\sigma)]\Big) \tag{C.27}$$

$$= \mathcal{O}\big(\sigma^{-1}\big)\mathcal{O}\big(\delta+\sigma\big) = \mathcal{O}\big(\sqrt{\log\epsilon^{-1}}\,\big) \tag{C.28}$$

where we have used $\delta \in \Theta\big(\sigma\sqrt{\log\epsilon^{-1}}\big)$. We therefore have $M_{\text{half}} \in e^{\mathcal{O}(\sqrt{\log\epsilon^{-1}})} \subset \mathcal{O}\big(\epsilon^{-1}\big)$ and thus also $M \leq M_{\text{half}}^2 \in \mathcal{O}(\epsilon^{-2}\,)$. We thus set $\ell \in \Theta\big(\frac{1}{\sqrt{\sigma}}\log(\epsilon^{-3}\sigma^{-1/2})\big)$ and compute

$$\Big| \mathcal{W}_{\sigma,\delta}(x) - [S_\ell \mathcal{W}_{\sigma,\delta}](x) \Big| \leq \frac{M}{r^{\ell+1}(r-1)} \tag{C.29}$$

$$\leq \frac{C\,\epsilon^{-2}}{(1+2\sqrt{\sigma})^{\frac{D}{\sqrt{\sigma}}\log(\epsilon^{-3}\sigma^{-1/2})}\, 2\sqrt{\sigma}} \tag{C.30}$$

$$\leq \frac{C\,\epsilon^{-2}}{e^{\log(\epsilon^{-3}\sigma^{-1/2})}\, 2\sqrt{\sigma}} = (\epsilon^3\,\sigma^{1/2})\,\frac{C\,\epsilon^{-2}}{2\sqrt{\sigma}} \tag{C.31}$$

$$\leq \epsilon/5 \tag{C.32}$$

where C and D are positive constants. We recall that $\sigma \in \Theta\big(\delta/\sqrt{\log(\epsilon^{-1})}\big)$ and thus

$$\ell \in \Theta\Big(\frac{1}{\sqrt{\sigma}}\big[3\log(\epsilon^{-1}) + \frac{1}{2}\log(\sigma^{-1})\big]\Big) \tag{C.33}$$

$$= \Theta\Big(\frac{1}{\sqrt{\delta}}\log^{1/4}(\epsilon^{-1})\big[\log(\epsilon^{-1}) + \log(\delta^{-1})\big]\Big) \tag{C.34}$$

is sufficient to guarantee that $[S_\ell \mathcal{W}_{\sigma,\delta}](x)$ is within $\epsilon/5$ distance from $\mathcal{W}_{\sigma,\delta}(x)$.

Finally, we renormalise $[S_\ell \mathcal{W}_{\sigma,\delta}](x)$ dividing it by the maximum value attained for $x \in [-1,1]$; by construction, this value is smaller than $1+\epsilon/5$, thus the maximum distance from $\mathcal{W}_{\sigma,\delta}$ after normalisation is bounded by $1 - \frac{1-\epsilon/5}{1+\epsilon/5} < \epsilon/2$.

□

C.3 Polynomial approximations on increasingly larger domains

Now we explain how the windowing functions can be used to select domain where a polynomial of low degree can be a good approximation of the function $1/(1-x)$, if it is applied to eigenvalues contained in such domain.

As usual, the spectrum of A is contained in $\mathcal{D}_A = \left[\frac{1}{\kappa}, 1\right]$ and, correspondingly, the spectrum of $B = I - \eta A$ is contained in $\mathcal{D}_B = \left[1-\eta, 1-\frac{\eta}{\kappa}\right] \subseteq \left[0, 1-\frac{\eta}{\kappa}\right]$. We set $m := \lceil \log_2 \kappa \rceil + 1$, $\delta_j := \eta\, 2^{-j}$ for $j = \{1, \dots, m\}$ and fix $\tilde{\epsilon} > 0$, a parameter related to the target precision ε. According to Eq. (25) and Eq. (C.13) we can find polynomials $P_{\tilde{\epsilon},\delta_j}(x)$ and $W_{\tilde{\epsilon},\delta_j}(x)$ such that

$$\left| P_{\tilde{\epsilon},\delta_j}(x) - \frac{1}{1-x} \right| \leq \tilde{\epsilon} \qquad \forall x \in [-1, 1-\delta_j] \tag{C.35}$$

$$W_{\tilde{\epsilon},\delta_j}(x) \qquad \text{satisfies Eq. (C.13) for each } \delta_j \tag{C.36}$$

with degrees in $\mathcal{O}\big(\delta^{-1/2} \log(\tilde{\epsilon}^{-1}\delta^{-1})\big)$ and $\mathcal{O}\big(\delta^{-1/2} \log^{1/4}(\tilde{\epsilon}^{-1}) \log(\tilde{\epsilon}^{-1}\delta^{-1})\big)$, respectively. We also define a normalisation factor

$$K := 2 \max_j \max_{x \in [-1,1]} |P_{\tilde{\epsilon},\delta_j}(x)| \tag{C.37}$$

which can be set to coincide with the factor K defined in Eq. (26) and introduce the shorthands

$$P_j(x) := P_{\tilde{\epsilon},\delta_j}(x)/K \quad \text{and} \quad W_j(x) := W_{\tilde{\epsilon},\delta_j}(x). \tag{C.38}$$

The windowing function $W_j(x)$ is used to select an interval $[-1+2\delta_j, 1-2\delta_j]$ where $P_j(x)$ is a good approximation of the inverse. For any eigenvalue λ of B we have $P_j(\lambda) \approx \frac{1}{K}\frac{1}{1-\lambda}$ when $\lambda \leq 1-\delta_j$ and $W_j(\lambda) \approx 0$ when $\lambda \geq 1-\delta_j$. More precisely, we have

$$W_j(B) P_j(B) = W_j(B) \frac{1}{K} \frac{1}{I-B} + \Delta_j^{\tilde{\epsilon}} \tag{C.39}$$

$$= W_j(B) \frac{A^{-1}}{\eta K} + \Delta_j^{\tilde{\epsilon}} \tag{C.40}$$

where $\Delta_j^{\tilde{\epsilon}}$ are arbitrary matrices having operator norm smaller or equal to $\tilde{\epsilon}$. Note that $W_j(B)$ and $P_j(B)$ commute, since they are both polynomial functions of B. Moreover, we can set without loss of generality $W_m(B) \equiv I$, since we already have $P_m(B) \approx A^{-1}/(\eta K)$ on the entire domain of A. Finally, we have $K \in \Theta(\kappa/\eta)$, as derived in the main text.

C.4 Variable-stopping-time PD-QLS solver: definition

We now proceed to reformulate our PD-QLS solver as a variable-stopping-time algorithm. For notation convenience, from now on we often drop the dependency of a polynomial from its variable and simply write P instead of $P(x)$.

Algorithm 27 (Variable-stopping-time linear system solving)**.** *We preliminarily define two variable-stopping-time algorithms $\mathcal{A}$ and $\mathcal{B}$. $\mathcal{A}$ is the core PD-QLS solving function and $\mathcal{B}$ is used to un-compute the clock registers at the end of the process.*

The algorithm $\mathcal{A} = \mathcal{A}_m \cdot \ldots \cdot \mathcal{A}_1 \cdot \mathcal{A}_0$, with $m := \lceil \log_2 \kappa \rceil + 1$, acts on the Hilbert space $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_F \otimes \mathcal{H}_S$ (respectively clock register, flag qubit, system register with $\dim \mathcal{H}_S = N$), plus ancillary registers if needed. We set $\mathcal{A}_0 = I_C \otimes I_F \otimes \mathcal{U}_{\boldsymbol{b}}$ (i.e., $\mathcal{A}_0$ prepares $|\boldsymbol{b}\rangle$ in register S) while the other sub-algorithms $\mathcal{A}_j$ use oracular access only to $\mathcal{U}_B$. Each $\mathcal{A}_j$, for $j \geq 1$, consists of the following two steps:

1. *Conditioned on the qubits $C_1, \dots, C_{j-1}$ being in $|0\rangle^{\otimes j-1}$, use QSP and a single Pauli X gate to implement the unitary*

$$\mathcal{U}'_j := \begin{pmatrix} \sqrt{1-W_j^2} & -W_j \\ W_j & \sqrt{1-W_j^2} \end{pmatrix} \quad \text{acting on } \mathcal{H}_{C_j} \otimes \mathcal{H}_S, \tag{C.41}$$

with the window function $W_j(B)$ given in Eq. (C.38).

2. *Conditioned on C_j being in $|1\rangle_{C_j}$, use QSP and a single Pauli X gate to implement the unitary*

$$\mathcal{U}''_j := \begin{pmatrix} \sqrt{I-P_j^2} & -P_j \\ P_j & \sqrt{I-P_j^2} \end{pmatrix} \quad \text{acting on } \mathcal{H}_F \otimes \mathcal{H}_S, \tag{C.42}$$

with the polynomial approximations $P_j(B)$ given in Eq. (C.38).

The algorithm $\mathcal{B} = \mathcal{B}_m \cdot \ldots \cdot \mathcal{B}_1 \cdot \mathcal{B}_0$ acts on the same Hilbert space as $\mathcal{A}$. The initialisation step is skipped, i.e., we set $\mathcal{B}_0 = I$, and $\forall j \geq 1$ we define $\mathcal{B}_j$ as the application of the unitary $\mathcal{U}'_j$ given in Eq. (C.41) controlled on $C_1, \dots, C_{j-1}$ being in $|0\rangle^{\otimes j-1}$. The unitaries $\mathcal{U}''_j$ are not applied.

The complete PD-QLS solving algorithm is then defined as follows.

1. *In a initial pre-processing step (that needs to be run only once), a sequence of integers $(k_0, k_1, \dots, k_m)$ that satisfy Eq. (C.10) is determined using a phase estimation algorithm.*

2. *The main part of the algorithm consists in running $\mathcal{A}'$, which is a VTAA version of $\mathcal{A}$ where each $\mathcal{A}'_j$ implements a k_j-step amplification of $\mathcal{A}_j \mathcal{A}'_{j-1}$.*

3. *The final post-processing consists in applying $\mathcal{B}^\dagger$ to the output of $\mathcal{A}'$ in order to un-compute the clock register. We then measure the flag qubit and when the result is $|1\rangle_F$ (which happens with constant probability) we output the S register (which contains a state close to $|A^{-1}\boldsymbol{b}\rangle$).*

C.5 Variable-stopping-time PD-QLS solver: correctness

We will now analyse the correctness of Algorithm 27 and we begin introducing a Lemma.

Lemma 28. *The state after the application of $\mathcal{A}_k$ defined in Algorithm 27 is given by*

$$\mathcal{A}_{\leq k} |0\rangle_{\text{all}} = \sum_{j=1}^{k} |1_j\rangle_C \left(|0\rangle_F M_{j-1} W_j \sqrt{I-P_j^2} |\boldsymbol{b}\rangle_S + |1\rangle_F M_{j-1} W_j P_j |\boldsymbol{b}\rangle_S \right) \tag{C.43}$$

$$+ M_k |0, 0, \boldsymbol{b}\rangle_{C,F,S} \tag{C.44}$$

with $M_0 := I$, $M_j := \prod_{i=1}^{j} \sqrt{I-W_i^2}$ and, using $W_m = I$, at the last step we have $M_m = 0$. Moreover, the algorithm $\mathcal{B}$ defined in Algorithm 27 acts as

$$\mathcal{B} |0, f, \phi\rangle_{C,F,S} = \sum_{j=1}^{k} |1_j\rangle_C |f\rangle_F M_{j-1} W_j |\phi\rangle_S \tag{C.45}$$

for any input state $|\phi\rangle_S$ and $f \in \{0, 1\}$.

Proof. We proceed by induction over k. For $k = 0$ we have $\mathcal{A}_{\leq 0} |0\rangle_{\text{all}} = \mathcal{A}_0 |0\rangle_{\text{all}} = |0, 0, \mathbf{b}\rangle_{C,F,S}$.

Using that P_j, M_j, W_j are functions of B and thus commute, we have at step $k+1$

$$\text{(C.44)} \overset{\mathcal{U}'_{k+1}}{\mapsto} \sum_{j=1}^{k} |1_j\rangle_C \left(|0\rangle_F M_{j-1} W_j \sqrt{I - P_j^2}\, |\mathbf{b}\rangle_S + |1\rangle_F M_{j-1} W_j P_j |\mathbf{b}\rangle_S \right) \tag{C.46}$$

$$+ |1_{k+1}\rangle_C |0\rangle_F M_k W_{k+1} |\mathbf{b}\rangle_F + |0,0\rangle_{C,F} M_k \sqrt{1 - W_{k+1}^2}\, |\mathbf{b}\rangle_{C,F,S} \tag{C.47}$$

$$\text{(C.47)} \overset{\mathcal{U}''_{k+1}}{\mapsto} \sum_{j=1}^{k} |1_j\rangle_C \left(|0\rangle_F M_{j-1} W_j \sqrt{I - P_j^2}\, |\mathbf{b}\rangle_S + |1\rangle_F M_{j-1} W_j P_j |\mathbf{b}\rangle_S \right) \tag{C.48}$$

$$+ |1_{k+1}\rangle_C \left(|0\rangle_F M_k W_{k+1} \sqrt{I - P_{k+1}^2}\, |\mathbf{b}\rangle_S + |1\rangle_F M_k W_{k+1} P_{k+1} |\mathbf{b}\rangle_S \right) \tag{C.49}$$

$$+ M_{k+1} |0,0,\mathbf{b}\rangle_{C,F,S} \,. \tag{C.50}$$

Equation (C.45) can be verified similarly. □

Next, we consider the output state of $\mathcal{A}'$, the VTAA version of $\mathcal{A}$, which features an amplification of the amplitude of the $|1\rangle_F$ component, i.e., it is a state of of the form

$$\mathcal{A}' |0\rangle_{\text{all}} = \sqrt{p'_{\text{succ}}}\, |1\rangle_F |\psi_{\text{succ}}\rangle_{C,S} + \sqrt{1 - p'_{\text{succ}}}\, |0\rangle_F |\psi_{\text{fail}}\rangle_{C,S} \tag{C.51}$$

where the success probability is constant, $p'_{\text{succ}} \in \Theta(1)$, and where we have

$$|\psi_{\text{succ}}\rangle_{C,S} = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=1}^{m} |1_j\rangle_C M_{j-1} W_j P_j |\mathbf{b}\rangle_S \tag{C.52}$$

$$\mathcal{N} = \sum_{j=1}^{m} \left|\left| M_{j-1} W_j P_j |\mathbf{b}\rangle \right|\right|^2 . \tag{C.53}$$

We now claim that, for an appropriate choice of the algorithm parameters, the error in $|\psi_{\text{succ}}\rangle_{C,S}$ is upper bounded by $\mathcal{O}(\varepsilon)$, and we will prove this claim in the next section. Thus we have:

$$|\psi_{\text{succ}}\rangle_{C,S} = \sum_{j=1}^{m} |1_j\rangle_C M_{j-1} W_j \left|A^{-1}\mathbf{b}\right\rangle_S + \mathcal{O}(\varepsilon) \tag{C.54}$$

where $\mathcal{O}(x)$ here denotes an arbitrary vector with 2-norm bounded by $\mathcal{O}(x)$. Note that Eq. (C.45) for $|\phi\rangle_S = \left|A^{-1}\mathbf{b}\right\rangle_S$ implies that $\sum_j |1_j\rangle_C M_{j-1} W_j \left|A^{-1}\mathbf{b}\right\rangle_S$ is a normalised state.

The final step of the PD-QLS algorithm consists in applying $\mathcal{B}^\dagger$ to the state in Eq. (C.51). Using again Eq. (C.45) we obtain:

$$\mathcal{B}^\dagger \mathcal{A}' |0\rangle_{\text{all}} = \sqrt{p'_{\text{succ}}}\, \left|0, 1, A^{-1}\mathbf{b}\right\rangle_{C,F,S} + \sqrt{1 - p'_{\text{succ}}}\, |0,0,\psi'\rangle_{C,F,S} + \mathcal{O}(\varepsilon) \,. \tag{C.55}$$

Measuring the flag in $|1\rangle_F$ then results with constant success probability in a vector that has $\mathcal{O}(\tilde{\epsilon})$ distance from the ideal output $\left|A^{-1}\mathbf{b}\right\rangle$.

C.6 Variable-stopping-time PD-QLS solver: error bound

In order to derive Eq. (C.54) we use Eq. (C.40) and write

$$|\psi_{\text{succ}}\rangle_{C,S} = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=1}^{m} |1_j\rangle_C M_{j-1} W_j \frac{A^{-1}}{\eta K} |\mathbf{b}\rangle_S + \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=1}^{m} |1_j\rangle_C \Delta_j^{\tilde{\epsilon}} |\mathbf{b}\rangle_S \tag{C.56}$$

$$= \frac{\sqrt{\widetilde{\mathcal{N}}}}{\sqrt{\mathcal{N}}} \sum_{j=1}^{m} |1_j\rangle_C M_{j-1} W_j \left|A^{-1}\mathbf{b}\right\rangle_S + \mathcal{O}(\tilde{\epsilon}\sqrt{m/\mathcal{N}}) \tag{C.57}$$

$$= \sum_{j=1}^{m} |1_j\rangle_C M_{j-1} W_j \left|A^{-1}\mathbf{b}\right\rangle_S + \mathcal{O}(\tilde{\epsilon}\sqrt{m/\mathcal{N}}) \tag{C.58}$$

and $\widetilde{\mathcal{N}}$ is defined below. To prove the last step, recall that $\sum_j |1_j\rangle_C M_{j-1}W_j \left|A^{-1}\mathbf{b}\right\rangle_S$ is a normalised state, hence Eq. (C.57) implies $\left|\sqrt{\widetilde{\mathcal{N}}/\mathcal{N}}-1\right| \in \mathcal{O}(\tilde{\epsilon}\sqrt{m/\mathcal{N}})$. Now we estimate $\mathcal{N}$ by using $\left|\sqrt{\widetilde{\mathcal{N}}}-\sqrt{\mathcal{N}}\right| \in \mathcal{O}(\tilde{\epsilon}\sqrt{m})$ and computing

$$\widetilde{\mathcal{N}} := \sum_{j=1}^{m} \left|\left| M_{j-1}W_j \frac{A^{-1}}{\eta\, K} |\mathbf{b}\rangle \right|\right|^2 \tag{C.59}$$

$$= \frac{||A^{-1}|\mathbf{b}\rangle||^2}{\eta^2 K^2} \sum_{j=1}^{m} \left|\left| M_{j-1}W_j \left|A^{-1}\mathbf{b}\right\rangle \right|\right|^2 \tag{C.60}$$

$$= \frac{||A^{-1}|\mathbf{b}\rangle||^2}{\eta^2 K^2}. \tag{C.61}$$

As proven in the main text, $K \in \mathcal{O}(\kappa/\eta)$. Thus we have $\widetilde{\mathcal{N}} \in \Omega(1/\kappa^2)$ and choosing

$$\tilde{\epsilon} \in \mathcal{O}\left(\frac{\varepsilon}{\kappa\sqrt{\log\kappa}}\right) \tag{C.62}$$

we also have $\mathcal{N} \in \Omega(1/\kappa^2)$ and $\tilde{\epsilon}\sqrt{m/\mathcal{N}} \in \mathcal{O}(\varepsilon)$. This condition is sufficient to guarantee that the output state $|\psi_{\text{succ}}\rangle$ is within ε-distance from the ideal output.

C.7 Variable-stopping-time PD-QLS solver: query complexity

We can now provide a upper bounds the query complexity of the VTAA algorithm using Eq. (C.12):

$$Q' \in \mathcal{O}\left(t_{\max}\sqrt{m} + \frac{t_{\text{avg}}}{\sqrt{p_{\text{succ}}}}\sqrt{m\,\log(t_{\max}/t_{\min})}\right).$$

We first compute the $\mathcal{U}_{\mathbf{b}}$-complexity. Since the non-amplified algorithm $\mathcal{A}$ accesses $\mathcal{U}_{\mathbf{b}}$ only in the first step, we have $t_{\min} = t_{\max} = t_{\text{avg}} = 1$. Using $m\,\tilde{\epsilon}^2 \in \mathcal{O}\big(\varepsilon^2/\kappa^2\big)$ we get

$$p_{\text{succ}} = \mathcal{N} \;\in\; \mathcal{O}\Big(\widetilde{\mathcal{N}} + m\,\tilde{\epsilon}^2\Big) = \mathcal{O}\left(\frac{||A^{-1}|\mathbf{b}\rangle||^2}{\kappa^2}\right) \tag{C.63}$$

and the using $m = \lceil \log_2 \kappa \rceil + 1$ we have

$$Q[\mathcal{U}_{\mathbf{b}}] \in \mathcal{O}\left(\sqrt{\log(\kappa)} + \frac{\kappa}{||A^{-1}|\mathbf{b}\rangle||}\right). \tag{C.64}$$

We now compute the $\mathcal{U}_B$-complexity. We have:

$$t_j = \sum_{i=1}^{j}[\deg(P_i)+\deg(W_i)] \;\in\; \mathcal{O}\Big(\delta_j^{-1/2}\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\delta_j^{-1})\Big) \tag{C.65}$$

$$t_{\max} = \sum_{i=1}^{m}[\deg(P_i)+\deg(W_i)] \;\in\; \mathcal{O}\Big(\sqrt{\kappa/\eta}\,\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\kappa/\eta)\Big) \tag{C.66}$$

$$t_{\min} = \quad \deg(P_1)+\deg(W_1) \;\in\; \mathcal{O}\Big(\sqrt{1/\eta}\,\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}/\eta)\Big). \tag{C.67}$$

Writing $|\mathbf{b}\rangle \equiv \sum_\lambda \beta_\lambda |\lambda\rangle$, where $|\lambda\rangle$ denotes an eigenvector of B relative to the eigenvalue λ and $\sum_\lambda |\beta_\lambda|^2 = 1$, the probability of stopping at time t_j is

$$p_j = \left|\left|\,\Pi_{C_j}\mathcal{A}_{\le j}\,|0\rangle_{\text{all}}\,\right|\right|^2 = \left|\left|\,M_{j-1}W_j\,|\mathbf{b}\rangle\,\right|\right|^2 \tag{C.68}$$

$$= \sum_\lambda |M_{j-1}(\lambda)W_j(\lambda)|^2\,|\beta_\lambda|^2 \tag{C.69}$$

$$\in\; \mathcal{O}\Big(\tilde{\epsilon} + \sum\nolimits_{\lambda\in(1-4\delta_j,1-\delta_j]}|\beta_\lambda|^2\Big). \tag{C.70}$$

Here, we have used $W_j(\lambda) \leq \tilde{\epsilon}$ for $\lambda \geq 1-\delta_j$, while for $\lambda < 1-2\delta_{j-1} = 1-4\delta_j$ we have

$$M_{j-1}(\lambda) \leq \sqrt{1-W_{j-1}^2(\lambda)} \leq \sqrt{1-(1-\tilde{\epsilon})^2} \leq \sqrt{2\tilde{\epsilon}}\,, \tag{C.71}$$

i.e., the expression $|M_{j-1}(\lambda)W_j(\lambda)|^2$ is non-negligible only for $\lambda \in (1-4\delta_j, 1-\delta_j]$. Now we estimate the ℓ_2-average runtime $t_{\text{avg}} = \sqrt{\sum_{j=1}^m p_j t_j^2}$ with

$$t_{\text{avg}}^2 \in \mathcal{O}\left(\sum_{j=1}^m \left[\tilde{\epsilon} + \sum\nolimits_{\lambda\in(1-4\delta_j,1-\delta_j]} |\beta_\lambda|^2\right]\left[\delta_j^{-1/2}\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\delta_j^{-1})\right]^2\right) \tag{C.72}$$

$$\subseteq \mathcal{O}\left(\tilde{\epsilon}\, t_{\max}^2 + \sum\nolimits_\lambda \frac{|\beta_\lambda|^2}{1-\lambda}\,\log^{1/2}(\tilde{\epsilon}^{-1})\log^2(\tilde{\epsilon}^{-1}\kappa/\eta)\right) \tag{C.73}$$

where we have used that for any positive function $f(\lambda)$

$$\sum_{j=1}^m \sum_{\lambda\in(1-4\delta_j,1-\delta_j]} f(\lambda)\,\frac{1}{\delta_j} \;\in\; \Theta\Big(\sum_\lambda f(\lambda)\,\frac{1}{1-\lambda}\Big). \tag{C.74}$$

We then write $\sum_\lambda \frac{|\beta_\lambda|^2}{1-\lambda} = \Big|\Big|\sum_\lambda \frac{\beta_\lambda}{\sqrt{1-B}}\,|\lambda\rangle\Big|\Big|^2 = \big|\big|(\eta\,A)^{-1/2}\,|\mathbf{b}\rangle\big|\big|^2$ and obtain

$$t_{\text{avg}} \in \mathcal{O}\left(\frac{1}{\sqrt{\eta}}\big|\big|A^{-1/2}\,|\mathbf{b}\rangle\,\big|\big|\,\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\kappa/\eta)\right) \tag{C.75}$$

where we used $\big|\big|A^{-1/2}\,|\mathbf{b}\rangle\big|\big| \geq 1$ and thus $\sqrt{\tilde{\epsilon}}\,t_{\max} \in \mathcal{O}\left(\frac{1}{\sqrt{\eta}\log^{1/4}\kappa}\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\kappa/\eta)\right)$ is a sub-leading contribution to t_{avg}. Then, using $1/\sqrt{p_{\text{succ}}} \in \mathcal{O}\big(\kappa/\big|\big|A^{-1}\,|\mathbf{b}\rangle\big|\big|\big)$ and

$$\log(t_{\max}/t_{\min}) \;\in\; \mathcal{O}\big(\log(\kappa)+\log\log(\tilde{\epsilon}^{-1}\kappa/\eta)\big) \;\subseteq\; \mathcal{O}\big(\log(\tilde{\epsilon}^{-1}\kappa/\eta)\big) \tag{C.76}$$

we finally obtain

$$Q[\mathcal{U}_B] \in \mathcal{O}\left(\sqrt{\frac{\kappa}{\eta}}\;\underbrace{\sqrt{\kappa}\frac{\big|\big|A^{-1/2}\,|\mathbf{b}\rangle\big|\big|}{\big|\big|A^{-1}\,|\mathbf{b}\rangle\,\big|\big|}}_{:=\Gamma_{A,\mathbf{b}}}\;\underbrace{\log^{1/4}(\tilde{\epsilon}^{-1})\log(\tilde{\epsilon}^{-1}\kappa/\eta)\sqrt{\log(\kappa)\log(\tilde{\epsilon}^{-1}\kappa/\eta)}}_{\text{polylog}(\kappa,\tilde{\epsilon}^{-1},\eta^{-1})}\right). \tag{C.77}$$

D Proof that the Sum-QLS problem is BQP-hard

In this Appendix we prove that the Sum-QLS problem is BQP-hard, therefore no efficient classical algorithm can solve the problem (unless BPP = BQP). The initial part of the proof is equal to the reduction presented by HHL: it is possible to construct a QLS problem $M\mathbf{x} = \mathbf{e}_1$ (where $\mathbf{e}_1$ is a canonical vector with a one in the first position) which is BQP-hard for a class of sparse indefinite matrices M that are easily constructible. This can be converted to the equivalent QLS problem $M\mathbf{x} = M^\dagger\mathbf{e}_1$, where the matrix $A = M^\dagger M$ is by construction PD and $\big|M^\dagger\mathbf{e}_1\big\rangle$ is easy to prepare. What remains to be proven is that A admits an explicit Sum-QLS structure, i.e. that one can construct a decomposition as a sum of PD local Hamiltonian terms and that the overlap with the support (bounded by the parameter γ) scales polynomially.

Preliminarily, we introduce a couple of useful definitions: the specific quantum circuit model (universal for quantum computation) that we aim to "simulate" as a Sum-QLS, and the Feynman-Kitaev clock construction.

Quantum circuit model: We describe an arbitrary quantum computation $\mathcal{C}$ on n qubits as the application of T elementary gates $\{U_0, U_1 \ldots, U_{T-1}\}$ to an initial state $|0\rangle^{\otimes n}$ and the output of the computation is the quantum state $U_{T-1}\cdots U_1 U_0\,|0\rangle^{\otimes n}$. We assume that the elementary gate set consists of gates acting either on one or two qubits, e.g. arbitrary single-qubit rotations and control-nots. Hence, the t-th elementary gate is described by a unitary $U_t \in \mathbb{C}^{N\times N}$ with $N = 2^n$ which can be written as $U_t = u_t \otimes I_{\neg\mathcal{S}_t}$, where u_t acts on a subset $\mathcal{S}_t$ of the qubits and is either a 2×2 ($|\mathcal{S}_t| = 1$) or a 4×4 ($|\mathcal{S}_t| = 2$) unitary matrix.

Feynman-Kitaev clock: We introduce the following unitary matrix acting on $\mathbb{C}^{3T} \otimes \mathbb{C}^n$, which is based on the *Feynman-Kitaev clock* construction (see e.g. [55] and references therein):

$$U := \sum_{t=0}^{3T-1} |t{+}1\rangle\langle t|_c \otimes U_t \tag{D.1}$$

$$= \sum_{t=0}^{T-1} |t{+}1\rangle\langle t|_c \otimes U_t \; + \; |T{+}t{+}1\rangle\langle T{+}t|_c \otimes I \; + \; |2T{+}t{+}1\rangle\langle 2T{+}t|_c \otimes U_{T-t-1}^\dagger \tag{D.2}$$

where we have defined $U_t = I$ for $T \leq t \leq 2T-1$ and $U_t = U_{3T-t-1}^\dagger$ for $2T \leq t \leq 3T-1$ and the sums in the clock register (denoted by the subscript c) are taken modulo $3T$.

Proposition 29 (Sum-QLS is BQP-hard)**.** *Let $\mathcal{C}$ be a given n-qubit T-gate quantum circuit. Then, there is a Sum-QLS problem (as in Definition 14) with*

1. $n' \in \mathcal{O}(n + \log T)$ *[$N' \in \mathcal{O}(2^{n'})$ is the size of A]*
2. $\kappa \in \mathcal{O}(T^2)$ *[condition number]*
3. $J \in \mathcal{O}(T)$ *[number of PD Hamiltonian terms]*
4. $s \in \mathcal{O}(\log T)$ *[locality of the Hamiltonian terms]*
5. $d_{\boldsymbol{b}} \in \mathcal{O}(1)$ *[sparsity of the known-term vector $\boldsymbol{b}$]*
6. $\gamma \in \Omega(T^{-2})$ *[overlap with the support, Eq. (91)]*

that is equivalent to $\mathcal{C}$, i.e., solving this Sum-QLS problem (up to a small constant error ε) allows to obtain the output state of $\mathcal{C}$ with constant probability and constant precision. According to Eq. (92), the algorithm presented in Section 5 can solve this Sum-QLS problem with a gate complexity in $\mathcal{O}\big(\mathrm{poly}(n, T)\big)$. When $T \in \mathcal{O}(\mathrm{poly}\, n)$, the gate complexity of the Sum-QLS solver is also in $\mathcal{O}(\mathrm{poly}\, n)$, i.e., Sum-QLS problem is BQP-hard. Moreover, the Sum-QLS$_{\mathrm{poly}}$, defined as the subclass of problems where the six parameters listed above all scale polynomially in n, is BQP-complete.

Proof. Starting from the Feynman-Kitaev unitary U encoding a quantum circuit $\mathcal{C}$ as in Eq. (D.1), we introduce the matrix

$$M := I - U\, e^{-1/T} \tag{D.3}$$

which can be written in block form (where each block has size $2^n \times 2^n$) as

$$M = \begin{pmatrix} \ddots & \ddots & 0 & & \\ \ddots & I & -U_t e^{-1/T} & 0 & \\ & 0 & I & -U_{t+1} e^{-1/T} & \\ & & 0 & I & \ddots \\ -U_{3T-1} e^{-1/T} & & & \ddots & \ddots \end{pmatrix} \tag{D.4}$$

and moreover we introduce:

$$A := M^\dagger M = (1 + e^{-2/T}) I - e^{-1/T}(U + U^\dagger) \tag{D.5}$$

where A is by construction Hermitian positive definite and the singular values of M are the square root of the eigenvalues of A. From the eigenvalue inequality $-2 \leq \lambda(U + U^\dagger) \leq 2$ we obtain

$$\begin{cases} \lambda_{\max}(A) \leq 4 \\ \lambda_{\min}(A) \geq \big(1 - e^{-1/T}\big)^2 \geq \frac{(1-e^{-1})^2}{T^2} \end{cases} \quad \Longrightarrow \quad \kappa(A) \leq \frac{4}{(1-e^{-1})^2}\, T^2 \tag{D.6}$$

hence $\kappa(M) = \sqrt{\kappa(A)} \in \mathcal{O}(T)$. Using the identity $U^{3T} = I$, the inverse of M can be expanded as

$$M^{-1} = \sum_{t'=0}^{\infty} U^{t'} e^{-t'/T} = \frac{e^3}{e^3 - 1} \sum_{t=0}^{3T-1} U^{t'} e^{-t'/T} \tag{D.7}$$

$$= \frac{e^3}{e^3 - 1} \sum_{t'=0}^{3T-1} e^{-t'/T} \sum_{t=0}^{3T-1} |t{+}t'\rangle\langle t| \otimes U_{[t:t+t']} \tag{D.8}$$

where the notation $U_{[a:b]}$ indicates the ordered product of unitary operators from a to $b-1$, $U_{[a:b]} := U_{b-1}U_{b-2}\cdots U_a$, with $U_{[t:t]} = I$.

As done already by HHL in Ref. [1] step, we consider the QLS problem

$$M\mathbf{x} = \mathbf{e}_1 \tag{D.9}$$

where $\mathbf{e}_1 \in \mathbb{C}^{3T2^n}$ is the unit vector with a one in the first position (i.e., it is the vector representation of the state $|0\rangle_c |0\rangle^{\otimes n}$). Using the state $|\mathbf{x}\rangle = \left|M^{-1}\mathbf{e}_1\right\rangle$ the output state of the quantum computation is obtained whenever upon measurement of the clock register a time $t \in [T : 2T-1]$ is returned, which happens with probability $e^{-2}/(1+e^{-2}+e^{-4}) \geq 0.11$. Therefore, any quantum circuit $\mathcal{C}$ having T gates can be restated as a QLS and solved (using e.g. the HHL algorithm) with a gate complexity scaling as $\mathrm{poly}(n, T)$. If the circuit $\mathcal{C}$ has $T \in \mathcal{O}(\mathrm{poly}\, n)$, then also the QLS solver has gate complexity in $\mathcal{O}(\mathrm{poly}\, n)$, showing that the QLS problem is BQP-hard.

We now consider the PD linear system

$$A\mathbf{x} = \mathbf{b} \qquad \text{for } A = M^\dagger M\,, \quad \mathbf{b} = M^\dagger \mathbf{e}_1 \tag{D.10}$$

which is equivalent to the system in Eq. (D.9) and can be cast as a Sum-QLS problem.

Notice, first, that $\mathbf{b}$ is a sparse vector, with $d_{\mathbf{b}} \leq 3$: if the elementary gate set consists of control-nots and single-qubit rotations, then the first row of M has at most 3 non-zero entries.

The matrix A has size $3T2^n \times 3T2^n$ and can be written as

$$A = \sum_{t=0}^{3T-1} H_{(t)}\,. \tag{D.11}$$

where each Hamiltonian terms $H_{(t)}$ is positive definite and only acts on the clock register plus either one or two extra qubits (i.e., the same number of qubits on which the gates from the elementary set act). We start writing A in block form

$$A = \begin{pmatrix} \ddots & \ddots & 0 & & \\ \ddots & I(1+e^{-2/T}) & -U_t e^{-1/T} & 0 & \\ 0 & -U_t^\dagger e^{-1/T} & I(1+e^{-2/T}) & -U_{t+1}e^{-1/T} & 0 \\ & 0 & -U_{t+1}^\dagger e^{-1/T} & I(1+e^{-2/T}) & \ddots \\ & & 0 & \ddots & \ddots \end{pmatrix} \tag{D.12}$$

and then introduce, for each $t \in \{0, \ldots, 3T-1\}$, the Hamiltonian term

$$H_{(t)} \;=\; I\,\delta \;+\; e^{-1/T} \left(\begin{array}{c|cc|c} 0 & 0 & 0 & 0 \\ \hline 0 & I & -U_t & 0 \\ 0 & -U_t^\dagger & I & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right) \begin{array}{l} \} \text{ size } t\cdot 2^n \\ \} \text{ size } 2\cdot 2^n \\ \} \text{ size } (3T-t-2)\cdot 2^n \end{array} \tag{D.13}$$

where we have defined

$$\delta \;:=\; \frac{1}{3T}\left(1+e^{-2/T}-2e^{-1/T}\right) \;\geq\; \frac{(1-e^{-1})^2}{3T^3} \;\geq\; \frac{1}{7.51\,T^3}\,, \tag{D.14}$$

and thus the decomposition in Eq. (D.12) holds. Note that the matrix $\left(\begin{smallmatrix} I & -U_t \\ -U_t^\dagger & I \end{smallmatrix}\right)$ has eigenvalues $\lambda = 0$ and $\lambda = 2$, hence is positive semi-definite, while each $H_{(t)}$ has the smallest eigenvalue equal to δ. Moreover, each $H_{(t)}$ is a local Hamiltonian. Specifically, we have $H_{(t)} = h_{(t)} \otimes I_{\neg \mathcal{S}_t}$, where $h_{(t)}$ acts on a set $\mathcal{S}_t$ consisting of $s = \lceil \log_2 3T \rceil + q$ qubits, where $q = 1$ if U_t corresponds to a single qubit gate and $q = 2$ if it corresponds to a two-qubit gate. Explicitly, we have

$$h_{(t)} \;=\; \left(\begin{array}{c|cc|c} I\,\delta & 0 & 0 & 0 \\ \hline 0 & I(e^{-1/T}+\delta) & -u_t e^{-1/T} & 0 \\ 0 & -u_t^\dagger e^{-1/T} & I(e^{-1/T}+\delta) & 0 \\ \hline 0 & 0 & 0 & I\,\delta \end{array}\right) \begin{array}{l} \} \text{ size } t\cdot 2^q \\ \} \text{ size } 2\cdot 2^q \\ \} \text{ size } (3T-t-2)\cdot 2^q \end{array} \tag{D.15}$$

where each u_t is a matrix specifying the t-th elementary quantum gate, with $U_t = u_t \otimes I_{\neg \mathcal{S}_t}$.

We now need to estimate a lower bound γ for the overlap parameter as in Eq. (91). We have

$$\langle \mathbf{b} | A^{-1} | \mathbf{b} \rangle = \left\langle M^\dagger \mathbf{e}_1 \right| M^{-1} M^{-\dagger} \left| M^\dagger \mathbf{e}_1 \right\rangle \tag{D.16}$$

$$= \frac{\langle \mathbf{e}_1 | M M^{-1} M^{-\dagger} M^\dagger | \mathbf{e}_1 \rangle}{||M^\dagger | \mathbf{e}_1 \rangle||^2} \tag{D.17}$$

$$= \frac{1}{||M^\dagger | \mathbf{e}_1 \rangle||^2} = \frac{1}{1 + e^{-2/T}} \geq 1/2 \tag{D.18}$$

while from (D.13) and (D.14) we obtain

$$\langle \mathbf{b} | H_{(j)}^{-1} | \mathbf{b} \rangle \leq \delta^{-1} \leq 7.51\, T^3 \,. \tag{D.19}$$

Using $J = 3T$ we finally get

$$\frac{1}{J^2} \frac{\sum_{j=1}^{J} \langle \mathbf{b} | H_{(j)}^{-1} | \mathbf{b} \rangle}{\langle \mathbf{b} | A^{-1} | \mathbf{b} \rangle} \leq \frac{1}{(3T)^2} \frac{3T \times 7.51\, T^3}{1/2} \leq 5.01\, T^2 =: \frac{1}{\gamma} \,. \tag{D.20}$$

In conclusion, given the sequence of gates u_t and the set of qubits $\mathcal{S}_t$ to which they are applied in the quantum circuit $\mathcal{C}$, we can efficiently compute the values and the positions of the non-zero entries of $H_{(t)}$, as given in Eq. (D.13). This then explicitly describes A as a sum of PD local Hamiltonian terms, as required by the definition of the Sum-QLS problem. Going through the derivation, we see that the relevant parameters scale as stated in points (1) to (6) in the statement of the Proposition. Using Eq. (92), it follows that the algorithm presented in Section 5 solves this problem with a gate complexity in $\mathcal{O}\big(\mathrm{poly}(n, T)\big) = \mathcal{O}(\mathrm{poly}\, n)$, in the case where the original circuit is itself a polynomial time quantum circuit (i.e. $T \in \mathcal{O}(\mathrm{poly}\, n)$). Finally, note that the ε-error in precision of Sum-QLS solver is amplified at most by a constant factor when post-selecting the clock register to show a time $t \in [T : 2T-1]$. Thus, it is possible to obtain the output of $\mathcal{C}$ with an error that is bounded by a constant, as required per the definition of the BQP class. This finally proves that the Sum-QLS problem is BQP-hard; at the same time, it proves that Sum-QLS$_{\mathrm{poly}}$, the sub-class of problems having polynomially scaling parameters, is solvable in quantum polynomial time and is thus BQP-complete.

□

References

[1] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, Physical Review Letters **103**, 150502 (2009) [arXiv:0811.3171].

[2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing, Third Edition*, Cambridge University Press (2007).

[3] Y. Saad, *Iterative methods for sparse linear systems*, SIAM (2003).

[4] A. Ambainis, *Variable time amplitude amplification and quantum algorithms for linear algebra problems*, 29th Symposium on Theoretical Aspects of Computer Science **14**, 636–647 (2012) [arXiv:1010.4458].

[5] A. M. Childs, R. Kothari, and R. D. Somma, *Quantum linear systems algorithm with exponentially improved dependence on precision*, SIAM J. Comput. **46**, 1920–1950 (2017) [arXiv:1511.02306].

[6] L. Wossnig, Z. Zhao, and A. Prakash, *Quantum linear system algorithm for dense matrices*, Physical Review Letters **120**, 050502 (2018) [arXiv:1704.06174].

[7] Y. Subaşı, R. D. Somma, and D. Orsucci. *Quantum algorithms for linear systems of equations inspired by adiabatic quantum computing*, Physical Review Letters **122**, 060504 (2019) [arXiv:1805.10549].

[8] Dong An and Lin Lin, *Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm*, arXiv:1909.05500 (2019).

[9] J. Wen, X. Kong, S. Wei, B. Wang, T. Xin, and G. Long, *Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing*, Physical Review A **99**, 012320 (2019) [arXiv:1806.03295].

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

[11] H. Y. Huang, K. Bharti, and P. Rebentrost, *Near-term quantum algorithms for linear systems of equations*, arXiv:1909.07344.

[12] L. Lin and Y. Tong, *Optimal quantum eigenstate filtering with application to solving quantum linear systems*, Quantum **4**, 361 (2020) [arXiv:1910.14596].

[13] S. Aaronson, *Read the fine print*, Nature Physics **11**, 291-293 (2015) [citeseerx].

[14] A. Montanaro and S. Pallister, *Quantum algorithms and the finite element method*, Physical Review A **93**, 032324 (2016) [arXiv:1512.05903].

[15] R. Babbush, J. McClean, C. Gidney, S. Boixo, and H. Neven, *Focus beyond quadratic speedups for error-corrected quantum advantage*, PRX Quantum **2** (2021) [arXiv:2011.04149].

[16] A. N. Chowdhury and R. D. Somma, *Quantum algorithms for Gibbs sampling and hitting-time estimation*, Quant. Inf. Comp. **17**, 0041–0064 (2017) [arXiv:1603.02940].

[17] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, *Preconditioned quantum linear system algorithm*, Physical Review Letters **110**, 250504 (2013) [arXiv:1301.2340].

[18] J. R. Shewchuk, *An introduction to the conjugate gradient method without the agonizing pain*, Carnegie Mellon University (1994).

[19] S. Chakraborty, A. Gilyén, and S. Jeffery, *The power of block-encoded matrix powers: improved regression techniques via faster hamiltonian simulation*, 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019) [arXiv:1804.01973].

[20] C. Shao and H. Xiang, *Quantum circulant preconditioner for linear system of equations*, Physical Review A **98**, 062321 [arXiv:1807.04563].

[21] Y. Tong, D. An, N. Wiebe, and L. Lin. *Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions*, Physical Review A **104**, 032422 [arXiv:2008.13295].

[22] B. Wu, M. Ray, L. Zhao, X. Sun, and P. Rebentrost, *Quantum-classical algorithms for skewed linear systems with optimized Hadamard test*, Physical Review A **103**, 042422 [arXiv:2009.13288].

[23] A. C. Vazquez, R. Hiptmair, and S. Woerner, *Enhancing the Quantum Linear Systems Algorithm using Richardson Extrapolation*, arXiv:2009.04484 (2020).

[24] G. H. Low and I. L. Chuang, *Optimal Hamiltonian simulation by quantum signal processing*, Physical Review Letters **118**, 010501 (2017) [arXiv:1606.02685].

[25] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, *Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics*, 51st Annual ACM SIGACT Symposium on Theory of Computing, 193–204 (2019) [arXiv:1806.01838].

[26] A. M. Childs and N. Wiebe, *Hamiltonian Simulation Using Linear Combinations of Unitary Operations*, Quantum Information & Computation [arXiv:1202.5822].

[27] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari and R. D. Somma, *Simulating Hamiltonian dynamics with a truncated Taylor series*, Physical Review Letters **114**, 090502 (2015) [arXiv:1412.4687].

[28] G. H. Low and I. L. Chuang, *Hamiltonian simulation by qubitization*, Quantum **3**, 163 (2019).

[29] A. C. Schaeffer, *Inequalities of A. Markoff and S. Bernstein for polynomials and related functions*, Bulletin of the American Mathematical Society **47**, 565–579(1941).

[30] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, *Quantum Amplitude Amplification and Estimation*, Contemporary Mathematics **305**, 53–74 (2002) [arXiv:0005055].

[31] See e.g. the Cholesky decomposition page on Wikipedia.

[32] R. D. Somma and S. Boixo, *Spectral gap amplification*, SIAM Journal on Computing **42**, 593-610 (2013) [arXiv:1110.2494].

[33] M. A. Nielsen, and I. Chuang, *Quantum computation and quantum information*, Cambridge University Press (2000).

[34] L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, arXiv:0208112 (2002).

[35] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum random access memory*, Physical Review Letters **100**, 160501 (2008) [arXiv:0708.1879].

[36] I. Kerenidis and A. Prakash, *Quantum recommendation systems*, arXiv:1603.08675.

[37] M. Boyer,G. Brassard, P. Høyer and A. Tapp, *Tight bounds on quantum searching*, Progress of Physics **46**, 493–505 (1998) [arXiv:9605034].

[38] András Gilyén, private communication.

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

[40] Y. Dong, X. Meng, K. B. Whaley and L. Lin, *Efficient phase factor evaluation in quantum signal processing*, Phys. Rev. A **103**, 042419 (2021) [arXiv:2002.11649].

[41] See e.g. the Gershgorin circle theorem page on Wikipedia.

[42] V. V. Shende, S. S. Bullock, and I. L. Markov, *Synthesis of quantum-logic circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**, 1000–1010 (2006) [arXiv:0406176].

[43] R. Merris, *Laplacian matrices of graphs: a survey*, Linear algebra and its applications **197**, 143–176 (1994).

[44] D. A. Spielman, *Algorithms, graph theory, and linear equations in Laplacian matrices*, Proceedings of the International Congress of Mathematicians 2010, 2698–2722 (2010).

[45] L. K. Grover, *Synthesis of quantum superpositions by quantum computation*, Physical Review Letters **85**, 1334 (2000).

[46] Y. R. Sanders, G. H. Low, A. Scherer and D. W. Berry, *Black-box quantum state preparation without arithmetic*, Physical Review Letters **122**, 020502 (2019) [arXiv:1807.03206].

[47] R. D. Somma and Y. Subaşı, *Quantum state verification in the quantum linear systems problem*, PRX Quantum **2**, 010315 (2021) [arXiv:2007.15698].

[48] A. Gilyén, *Quantum walk based search methods and algorithmic applications*, Doctoral dissertation, Eötvös Loránd University (2014).

[49] E. Malvetti, R. Iten, and R. Colbeck, *Quantum Circuits for Sparse Isometries* arXiv:2006.00016 (2020).

[50] X. Jiang, *Minimum rank positive semidefinite matrix completion with chordal sparsity pattern*, Doctoral dissertation, UCLA (2017).

[51] A. Nayak and F. Wu, *The quantum query complexity of approximating the median and related statistics*, Proceedings of the 31st annual ACM symposium on Theory of computing, 384–393 (1999) [arXiv:9804066].

[52] S. U. Pillai, T. Suel, and S. Cha, *The Perron-Frobenius theorem: some of its applications*, IEEE Signal Processing Magazine **22**, 62–75 (2005).

[53] S. Arora and B. Barak, *Computational complexity: a modern approach*, Cambridge University Press (2009).

[54] J. C. Mason, and D. C. Handscomb, *Chebyshev polynomials*, CRC press (2002).

[55] J. Bausch and E. Crosson, *Analysis and limitations of modified circuit-to-Hamiltonian constructions*, Quantum **2**, 94 (2018).