# Error convergence of quantum linear system solvers

Matias Ginzburg [*]
*SISSA, Trieste, Italy*

Ugo Marzolino [†]
*University of Trieste, Trieste, Italy*
*and Istituto Nazionale di Fisica Nucleare, Trieste Unit, Italy*

(Received 14 January 2026; accepted 19 May 2026; published 12 June 2026)

The Harrow-Hassidim-Lloyd (HHL) algorithm remains a paradigmatic quantum routine for solving linear systems and a widely used benchmark in theoretical and experimental studies. Several circuit-level variants have been proposed to simplify its implementation, including constructions in which the clock register is factorized via Hadamard gates to reduce circuit depth and entanglement. In this work, we show that this commonly deployed modification can fundamentally alter the asymptotic behavior of the algorithm and may fail to converge in the limit of large clock size due to persistent phase interference effects. This phenomenon is not a finite-size artifact and compromises the correctness guarantees of the factorized-clock implementation. We provide an analytic characterization of the underlying mechanism and demonstrate that a minimal and experimentally feasible modification suffices to restore convergence while preserving the practical advantages of the simplified circuit. Our results clarify an overlooked correctness issue in a widely used HHL implementation pattern and delineate the conditions under which simplified realizations faithfully reproduce the intended algorithmic behavior.

DOI: 10.1103/rjvs-sj7h

## I. INTRODUCTION

Quantum computers exploit the laws of quantum mechanics to perform computations in exponentially large Hilbert spaces, enabling potential speedups over classical algorithms [1,2]. It is conjectured that quantum algorithms can solve in polynomial time (with respect to the problem size) problems than cannot be solved polynomially by classical algorithms, like the factorization problem [3], and even problems whose solutions are not thought to be polynomially verifiable by classical computers, like the boson sampling problem [4,5] (namely outside the so-called NP class), but cannot probably solve NP-complete problems, i.e., the hardest problems in NP [6,7].

Among computational problems that are ubiquitous in computer science and big data [8,9] and attract attention in quantum computation, there are solutions of linear systems. The inputs of a linear problem are a vector $\mathbf{b} \in \mathbb{C}^d$ with components $b_j$, $j = 0, \dots d-1$ and a matrix $A \in \mathbb{C}^{d\times d}$ with components $A_{i,j}$, eigenvectors $\mathbf{u}_j$, and eigenvalues $\lambda_j$, $i, j = 0, \dots d-1$. The output is a vector $\mathbf{x} \in \mathbb{C}^d$ satisfying

$$A \cdot \mathbf{x} = \mathbf{b}. \tag{1}$$

[*]Contact author: mginzbur@sissa.it
[†]Contact author: ugo.marzolino@ts.infn.it



We focus on $s$-sparse matrices, namely matrices with a maximum number $s$ of nonvanishing elements in each row or column. Furthermore, we assume the matrix is Hermitian and define the condition number as $\kappa = \max_j |\lambda_j| / \min_j |\lambda_j|$. In Sec. II, we discuss how to modify the algorithm when the matrix is not Hermitian. The other relevant parameter is $\epsilon$, defined as the tolerated error of the output.

The first quantum algorithm for linear systems was proposed by Harrow, Hassidim, and Lloyd [10] (henceforth the *HHL algorithm*), although more recent algorithms have been studied and reviewed in [11]. Algorithms based on the adiabatic theorem [12,13] and eigenstate filtering [14] reach the optimal scaling $O(\kappa \ln(1/\epsilon))$ for the number of queries to matrix elements of $A$. Other algorithms, relying on quantum singular value transformation [15,16], exhibit the scaling $O(\kappa^2 \ln(\kappa/\epsilon))$. These recent algorithms employ the so-called block encoding technique that may require large time or memory [16–18]. Therefore, although the HHL algorithm shows a nonoptimal scaling $O(\kappa^2/\epsilon)$, it remains relevant not only for its conceptual simplicity but also for the simpler implementation with noisy intermediate-scale quantum (NISQ) technologies [19,20] compared to other algorithms.

The HHL algorithm is based on two protocols, the quantum phase estimation [1,21–23] and the amplitude amplification [24] (a variant of Grover's search algorithm [25–27]), that represent the core of general quantum algorithms (solving the so-called BQP-complete problems) [28]. In the original algorithm, the quantum phase estimation is initialized with a complex state that minimizes the final error. A variant of the HHL algorithm employing a much simpler initial state (i.e., prepared using Hadamard gates) is often discussed in the literature as it enables experimental implementations with currently available quantum hardware [29–33]. In the following, we refer to this version of the HHL algorithm as the *HHL variant*.

In both versions of the algorithm, there are two main registers; the first register encodes the vector $\mathbf{b}$ and the second is used to store the eigenvalues of the matrix $A$. Increasing the number of qubits in the second register enhances the precision of the eigenvalue estimation and reduces the error of the final output. Since the HHL variant assumes a known upper bound on the condition number, we extend the analysis to the case where such a bound is unknown, showing that the error fails to converge to zero even for a large number of qubits. Algorithms for estimating the condition number of an arbitrary sparse matrix are based on the Lanczos algorithm [34], the lower-upper decomposition, or the $QR$ decomposition of the matrix $A$. However, the running time of known algorithms [35–37] has the same scaling with respect to the parameters $s$ and $d$ as classical algorithms that solve linear problems, namely $O(sd)$ times the number of iterations required in specific algorithms. Therefore, the extension of the HHL algorithm to the case of unknown condition number is relevant as its computation typically negates the advantage of using quantum linear solvers.

Our analysis shows that this behavior is not a finite-size artifact but a structural feature of the algorithm, which can lead to misleading conclusions if left unaccounted for in theoretical analyses or experimental demonstrations. We then propose a modification of the HHL variant that preserves its simplicity, thus remaining experimentally feasible, and restores the convergence of the final output error to zero for large number of qubits in the second register.

We also carefully consider the complexity of the algorithm in the light of recent results on digital [38–47] or analog [48–51] simulations of unitaries used in the quantum phase estimation step. This analysis is the basis for the identification of problem instances in which HHL algorithms are competitive with more recent quantum linear solvers [11], which exhibit a more favorable complexity scaling with the computational error.

In Sec. II, we review the original HHL algorithm and its variant. In Sec. III, we analyze the convergence of the algorithm outcome towards the exact solution, and emphasize the difference between the original and the variant algorithms. The improved HHL variant is derived in Sec. IV. We compare the complexity of the above algorithms in Secs. V and VI. Sections VII and VIII are devoted, respectively, to numerical simulations of the HHL variant and the improved algorithm, and to the relaxation of a technical assumption (the positivity of $A$). Section IX summarizes our conclusions, while detailed derivations are provided in the Appendixes.

## II. HHL ALGORITHM

The linear problem (1) has a solution if and only if $\mathbf{b}$ lies in the support of $A$. In this case, given the eigenvectors $\mathbf{u}_j$ of $A$ with eigenvalues $\lambda_j \neq 0$, there is a decomposition of the input vector $\mathbf{b} = \sum_j \beta_j \mathbf{u}_j$ for some $\beta_j \in \mathbb{C}$ and the solution is $\mathbf{x} = \sum_j (\beta_j/\lambda_j)\mathbf{u}_j$. An alternative decomposition of the input state is $\mathbf{b} = \sum_j b_j \mathbf{e}_j$, where $\mathbf{e}_j$ are basis vectors with elements $(\mathbf{e}_j)_l = \delta_{j,l}$.

For later convenience, assume that $\mathbf{b}$ has unit $L^2$-norm, $\|\mathbf{b}\|_2 = 1$. If this condition is not met, the linear problem (1) can be mapped to an equivalent problem by dividing $\mathbf{b}$ and $\mathbf{x}$ by the norm $\|\mathbf{b}\|_2$. Assume also that $A$ is Hermitian, as the problem with non-Hermitian $A$ can be embedded in $\mathbb{C}^{2d}$ with a Hermitian matrix. Consider indeed the following problem:

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \cdot \mathbf{y} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}, \tag{2}$$

whose solution is the solution of the original problem embedded in $\mathbb{C}^{2d}$: $\mathbf{y} = \mathbf{0} \oplus A^{-1}\mathbf{b} = \mathbf{0} \oplus \mathbf{x}$ where $\mathbf{0} \in \mathbb{C}^d$ is the vector with all components set to zero. Solving the new problem then solves the original problem at the cost of doubling the dimension of the solution space. This requires only one extra qubit and the condition number of this new matrix is the ratio of the maximum and minimum singular value of $A$. Finally, assume without loss of generality that the eigenvalues of $A$ satisfy $0 < \lambda_j \leqslant 1$. The conditions $\lambda_j \leqslant 1$ can be achieved by dividing $A$ and multiplying $\mathbf{x}$ by a rough estimate (rounded up) of the largest eigenvalue of $A$. We will show in Sec. VIII how to relax the condition $\lambda_j > 0$.

The HHL algorithm and its variant use three quantum registers.

(1) The register $r$ consists of $n = \lceil \log_2 d \rceil$ qubits used to encode the vector $\mathbf{b}$ (padded with zeros if needed) in the pure state $|b\rangle_r = \sum_{j=0}^{d-1} b_j |j\rangle_r$, where $\{|j\rangle_r\}_{j=0,\dots,2^n-1}$ is the computational basis, i.e. the eigenvectors of the Pauli matrix $\sigma_z$ for each qubit. This is the reason for requiring $1 = \|\mathbf{b}\|_2^2 = \langle b|b\rangle$. This pure state can also be expressed as $|b\rangle_r = \sum_j \beta_j |u_j\rangle_r$, where $|u_j\rangle_r = \sum_k V_{k,j} |j\rangle_r$ and $V = [V_{k,j}]$ is the unitary matrix that diagonalizes $A$, $V^\dagger A V = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$. The space $\mathbb{C}^d$, where the vetors $\mathbf{x}$ and $\mathbf{b}$ lie, is then encoded in a subspace of the Hilbert space of the register $r$, that is the span of the $n$-qubit states $\{|j\rangle_r\}_{j=0,\dots,2^n-1}$ and has dimension $2^n \geqslant d$ compatibly with $n = \lceil \log_2 d \rceil$.

(2) The register $c$, called the *clock*, is used to implement the quantum phase estimation and to store the binary representation of eigenvalues of $A$ as elements of the computational basis. It consists of $n_c$ qubits, so that the states in the computational basis represent up to $T = 2^{n_c}$ different values and consequently the maximum precision for eigenvalue estimations is $2^{-n_c} = T^{-1}$. When $n_c$ and $T$ increase, the eigenvalues of $A$ are discriminated with better precision leading to a smaller error in the final output. For this reason, we use $T$ as the parameter to analyze the convergence towards the exact solution.

(3) The algorithm also uses an additional ancillary qubit $a$.

The solution $\mathbf{x}$ is encoded in the amplitudes of the output state vector of the $r$ register. Nevertheless, it is impossible to generate a state whose amplitudes are the components of $\mathbf{x}$, namely $\sum_j x_j |u_j\rangle_r$ with $x_j = \beta_j/\lambda_j$, because it is not normalized in general. Instead, the HHL algorithm aims at producing a state as close as possible to $\sum_j x_j |u_j\rangle_r / \|\mathbf{x}\|_2$. The amplitudes can be precisely measured with state tomography which is unfortunately not efficient in the worst case scenario

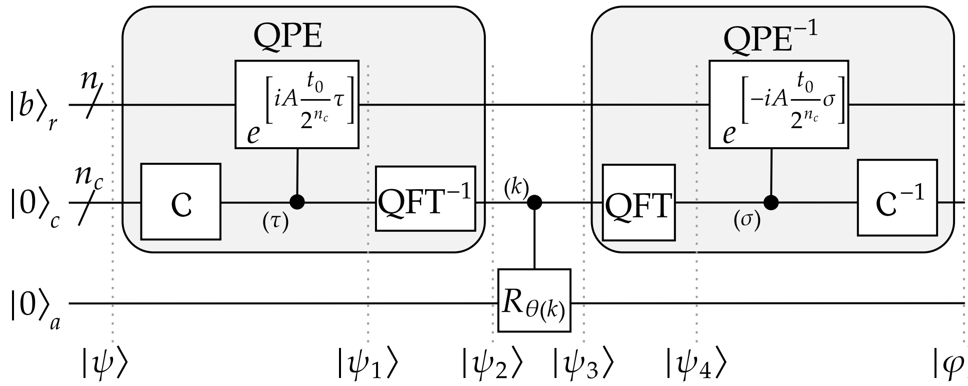


FIG. 1. Circuit of the main part of the HHL algorithm and of its variant. The first three operators implement the phase estimation protocol. The operation $\mathcal{C}$ represents the preparation of the *clock*; it can be chosen as in (4) to implement the original algorithm or as in (5) to implement its variant. The notations $(\tau)$, $(k)$, and $(\sigma)$ indicate controlled operations conditioned on the computational basis states $|\tau\rangle_c$, $|k\rangle_c$, and $|\sigma\rangle_c$ of the $c$ register respectively. The angle of the rotation $R_{\theta(k)}$ is defined by $\sin\theta(k) := Ct_0/(2\pi k)$. The last three steps uncompute the phase estimation.

[52–55]. Therefore, the HHL algorithm does not compute the exact solution $\mathbf{x}$, but it can be used to compute the norm $\|\mathbf{x}\|_2$ and expectation values $\mathbf{x}^\dagger M\mathbf{x}$ of matrices $M$ that can be implemented efficiently on the quantum register.

The main part of the original HHL algorithm [10] and of its variant [29–33,56] is compactly described by the circuit in Fig. 1. The explicit formulas of intermediate states are reported in Appendix A. The quantum registers are initially prepared in the following state:

$$|\psi\rangle = |0\rangle_a|0\rangle_c\sum_j \beta_j|u_j\rangle_r. \tag{3}$$

The difference between the HHL algorithm and its variant is the first operation on the clock, that is respectively

$$\mathcal{C}^{\text{HHL}}|0\rangle_c = \sqrt{\frac{2}{T}}\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi(2\tau+1)}{2T}\right)|\tau\rangle_c, \tag{4}$$

$$\mathcal{C}^{\text{var}}|0\rangle_c = \frac{1}{\sqrt{T}}\sum_{\tau=0}^{T-1}|\tau\rangle_c. \tag{5}$$

In the HHL variant, the operator $\mathcal{C}^{\text{var}}$ is the Hadamard gate on each qubit of the clock $|\gamma\rangle_c \to H^{\otimes n_c}|\gamma\rangle_c = \frac{1}{\sqrt{T}}\sum_{\sigma=0}^{T-1}\exp\left(i\pi\sum_{l=1}^{n_c}\gamma_l\sigma_l\right)|\sigma\rangle_c$ with $\{\gamma_l\}_l$ and $\{\sigma_l\}_l$ the binary representations of $\gamma$ and $\sigma$. Therefore, these Hadamard gates can be parallelized such that they require constant time, and the state $\mathcal{C}^{\text{var}}|0\rangle_c$ is fully factorized. The operator $\mathcal{C}^{\text{HHL}}$ in the HHL algorithm is more complicated as it entangles the clock qubits, and can be implemented in time $O(n_c) = O(\log_2 T)$ [57,58]. Although the variant slightly reduces circuit complexity, this simplification significantly lowers the precision of the eigenvalue estimation. The initial state of the $r$ register can be prepared following known state preparation schemes [57–68], or may result from a previous quantum protocol.

In Fig. 1, we have denoted by QFT the quantum Fourier transform $|\tau\rangle_c \to \frac{1}{\sqrt{T}}\sum_{k=0}^{T-1}e^{-i2\pi\tau k/T}|k\rangle_c$, and by $\text{QFT}^{-1}$ its inverse. The other gates are controlled unitaries defined by the following state transformations:

$$\begin{aligned}|\tau\rangle_c|b\rangle_r &\longrightarrow |\tau\rangle_c\, e^{iA\frac{t_0}{T}\tau}|b\rangle_r\\ &= \left|\tau_1,\tau_2,\ldots,\tau_{n_c}\right\rangle_c\prod_{j=1}^{n_c}e^{iA\frac{t_0}{T}2^{j-1}\tau_j}|b\rangle_r,\end{aligned} \tag{6}$$

$$|0\rangle_a|k\rangle_c \longrightarrow \begin{cases}\sin[\theta(k)]\,|1\rangle_a|k\rangle_c + \cos[\theta(k)]\,|0\rangle_a|k\rangle_c & \text{if } k \geqslant k_{\min}\\ |0\rangle_a|k\rangle_c & \text{if } k < k_{\min}\end{cases}, \tag{7}$$

where $\{\tau_j\}_j$ the binary representations of $\tau$, $t_0$, and $k_{\min}$ are free parameters and $\sin[\theta(k)] := Ct_0/(2\pi k)$ with $0 < C \leqslant 2\pi k_{\min}/t_0$. The implementability of the transformation (6) on a quantum computer requires the Hermiticity of $A$ assumed above.

The state vector after these operations for both algorithms is of the form

$$\begin{aligned}|\varphi\rangle = &\frac{1}{\sqrt{T}}\sum_{j=0}^{2^n-1}\sum_{\sigma=0}^{T-1}\sum_{k=k_{\min}}^{T-1}\beta_j\,\alpha_{k|j}\,\sin[\theta(k)]\,e^{i\frac{\sigma}{T}(2\pi k-\lambda_j t_0)}|1\rangle_a\,\mathcal{C}^{-1}|\sigma\rangle_c|u_j\rangle_r\\ &+\frac{1}{\sqrt{T}}\sum_{j=0}^{2^n-1}\sum_{\sigma=0}^{T-1}\sum_{k=k_{\min}}^{T-1}\beta_j\,\alpha_{k|j}\,\cos[\theta(k)]\,e^{i\frac{\sigma}{T}(2\pi k-\lambda_j t_0)}|0\rangle_a\,\mathcal{C}^{-1}|\sigma\rangle_c|u_j\rangle_r\\ &+\frac{1}{\sqrt{T}}\sum_{j=0}^{2^n-1}\sum_{\sigma=0}^{T-1}\sum_{k=0}^{k_{\min}-1}\beta_j\,\alpha_{0|j}\,e^{i\frac{\sigma}{T}(2\pi k-\lambda_j t_0)}|0\rangle_a\,\mathcal{C}^{-1}|\sigma\rangle_c|u_j\rangle_r,\end{aligned} \tag{8}$$

with coefficients

$$\alpha_{k|j}^{\mathrm{HHL}} = \frac{\sqrt{2}}{T}\sum_{\tau=0}^{T-1} e^{i\frac{\tau}{T}(\lambda_j t_0 - 2\pi k)} \sin\left(\frac{\pi(2\tau+1)}{2T}\right) = e^{i\phi}\frac{\sqrt{2}\cos\left[\frac{1}{2}(\lambda_j t_0 - 2\pi k)\right]\cos\left[\frac{1}{2T}(\lambda_j t_0 - 2\pi k)\right]\sin\left(\frac{\pi}{2T}\right)}{T\sin\left[\frac{1}{2T}(\lambda_j t_0 - 2\pi k+\pi)\right]\sin\left[\frac{1}{2T}(\lambda_j t_0 - 2\pi k-\pi)\right]}, \tag{9}$$

$$\alpha_{k|j}^{\mathrm{var}} = \frac{1}{T}\sum_{\tau=0}^{T-1} e^{i\frac{\tau}{T}(\lambda_j t_0 - 2\pi k)} = e^{i\phi}\frac{\sin\left[\frac{1}{2}(\lambda_j t_0 - 2\pi k)\right]}{T\sin\left[\frac{1}{2T}(\lambda_j t_0 - 2\pi k)\right]}, \tag{10}$$

for the original HHL algorithm and its variant. The phase $\phi = \frac{1}{2}(\lambda_j t_0 - 2\pi k)(1-\frac{1}{T})$ is not relevant in the following analysis.

The first three operators in Fig. 1 for the HHL variant are the standard quantum phase estimation algorithm used to estimate the eigenvalues of $A$. The operation $\mathcal{C}^{\mathrm{HHL}}$ in the original HHL algorithm prepares a state that minimizes the error of the quantum phase estimation [69–71]. The controlled rotations defined in (7) implement the inversion of the eigenvalues, requiring the register $a$ to be in the state $|1\rangle_a$ when the inversion succeeds and in $|0\rangle_a$ otherwise. The parameter $k_{\min}$ acts as a cutoff on the spectrum of $A$. The register state $|k\rangle_c$ encodes the eigenvalue $2\pi k/t_0$, such that the operator (7) correctly inverts the eigenvalues bigger than $2\pi k_{\min}/t_0$. Setting the minimum possible value $k_{\min}=1$ allows us to invert the largest number of eigenvalues, as is typically done in applications of the HHL variant [29–33,56]. If an upper bound of the condition number $\kappa \leqslant \kappa'$ is available, then we can set $k_{\min} = \lfloor t_0/(4\pi\kappa')\rfloor$. Since the coefficients $\alpha_{k|j}$ are centered around $2\pi k/t_0 \approx \lambda_j$ and recalling $\sin\theta(k) := Ct_0/(2\pi k)$, the coefficients in the first line of Eq. (8) are estimates of the desired coefficients $x_j = \beta_j/\lambda_j$.

The outputs of both the original HHL algorithm and its variant are then achieved by measuring the ancillary qubit in the basis $\{|0\rangle_a, |1\rangle_a\}$ and postselecting the outcome 1. This outcome occurs with probability

$$p_0 = \sum_{j=0}^{2^n-1} |\beta_j|^2 \sum_{k=k_{\min}}^{T-1} |\alpha_{k|j}|^2 \left(\frac{C\,t_0}{2\pi k}\right)^2, \tag{11}$$

which can be small. Nevertheless, one can find the outcome $|1\rangle_a$ after applying the amplitude amplification algorithm [24,27] to the state (8) with an average running time $O(1/\sqrt{p_0})$. The final normalized outputs, after the circuits in Fig. 1, the amplitude amplification, and the postselection, are of the form

$$|x_0\rangle = \frac{1}{\sqrt{T\,p_0}}\sum_{j=0}^{2^n-1}\sum_{\sigma=0}^{T-1}\sum_{k=k_{\min}}^{T-1} \beta_j\,\alpha_{k|j} \times \frac{C\,t_0}{2\pi k}\, e^{i\frac{\sigma}{T}(2\pi k - \lambda_j t_0)} |1\rangle_a \mathcal{C}^{-1}|\sigma\rangle_c |u_j\rangle_r, \tag{12}$$

with the coefficients (9) or (10).

## III. ERROR ANALYSIS

In this section, we discuss how close the output states of the HHL algorithm and of its variant are to the ideal solution of the linear problem (1). The error analysis presented below is independent of specific problem instances and relies only on the structure of the algorithm and on standard assumptions about the spectral properties of the input matrix.

The ideal output of the algorithm is the embedding of the vector $\mathbf{x}$ in the $r$ register as follows:

$$|x\rangle = |1\rangle_a |0\rangle_c \frac{C}{\sqrt{p}}\sum_{j=0}^{2^n-1}\frac{\beta_j}{\lambda_j}|u_j\rangle_r, \tag{13}$$

where we wrote the normalization constant as $C/\sqrt{p}$ in order to make the expression more similar to (12), and the normalization of the state (13) entails

$$p = \sum_{j=0}^{2^n-1}\left(\frac{C\,|\beta_j|}{\lambda_j}\right)^2. \tag{14}$$

Since $\sin[\theta(k)] = Ct_0/(2\pi k)$ is an amplitude of the state in (7), the constant $C$ satisfies $C \leqslant 2\pi k/t_0$ for all $k \geqslant k_{\min}$. We then choose the largest of the allowed values $C = 2\pi k_{\min}/t_0 \simeq 1/(2\kappa')$ that maximizes the ideal and the actual probability of measuring $|1\rangle_a$, respectively $p$ and $p_0$.

Before proceeding with our analysis, we stress that the precision of the eigenvalue estimation in the phase estimation subroutine is improved by increasing $t_0$. The reason is that the eigenvalues $\lambda_j$ are amplified and can be better distinguished. On the other hand, the phase estimation cannot differentiate phases modulo $2\pi$, and if $\lambda_j t_0/T > 2\pi$ the protocol fails. Therefore, under the assumption $0 < \lambda_j \leqslant 1$ for all $j$, we can take $t_0$ proportional to $T$, $t_0 = tT$ for some $0 < t < 2\pi$. This scaling implies $p = O(C/\lambda_{\min})^2 = O[(2\pi k_{\min})^2/(\lambda_{\min} tT)^2] = O(\kappa/\kappa')^2$.

The states $|x\rangle$ and $|x_0\rangle$ are close to each other if their squared overlap, also called fidelity $|\langle x|x_0\rangle|^2$, is close to 1. We then consider the so-called infidelity $1-|\langle x|x_0\rangle|^2$ as a measure of the deviation of the output state $|x_0\rangle$ from the ideal state $|x\rangle$. Indeed, the square root of the infidelity satisfies the properties of a distance between pure states $d(|x\rangle,|x_0\rangle) = \sqrt{1-|\langle x|x_0\rangle|^2}$ [72,73], and we shall use $d(|x\rangle,|x_0\rangle)$ to quantify the error of the algorithm. A straightforward computation gives

$$\langle x|x_0\rangle = \frac{C^2}{\sqrt{p\,p_0}}\sum_{j=0}^{2^n-1}\frac{|\beta_j|^2}{\lambda_j}\sum_{k=k_{\min}}^{T-1}|\alpha_{k|j}|^2\frac{t\,T}{2\pi k} = \frac{\sum_{j=0}^{2^n-1}\frac{|\beta_j|^2}{\lambda_j}\sum_{k=k_{\min}}^{T-1}|\alpha_{k|j}|^2\frac{t\,T}{2\pi k}}{\sqrt{\sum_{l=0}^{2^n-1}\frac{|\beta_l|^2}{\lambda_l^2}\sum_{j=0}^{2^n-1}|\beta_j|^2\sum_{k=k_{\min}}^{T-1}\left(|\alpha_{k|j}|\frac{t\,T}{2\pi k}\right)^2}}. \tag{15}$$

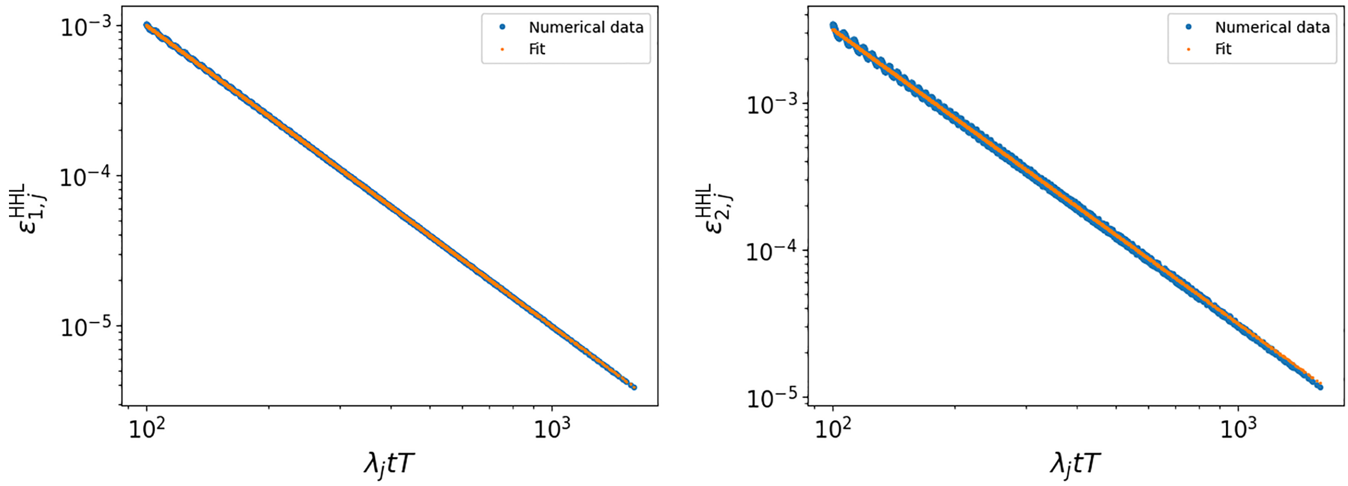


FIG. 2. Numerical computations of $\epsilon_{1,j}$ (left panel) and $\epsilon_{2,j}$ (right panel) for 50 equally spaced values of $\lambda_j \in (0, 1)$, 50 equally spaced values of $t \in (0.1\,\pi, \pi)$, and $n_c$ ranging from 3 to 9 ($T \in [2^3, 2^9]$). The orange curves are fits with functions $a_1(\lambda_j tT)^{-2}$ (left panel) and $a_2(\lambda_j tT)^{-2}$ (right panel) which result in $a_1 \simeq 9.94$ and $a_2 \simeq 31.54$.

Moreover, the following quantities,

$$\epsilon_{1,j} = \lambda_j \left( \sum_{k=k_{\min}}^{T-1} |\alpha_{k|j}|^2 \frac{tT}{2\pi k} - \frac{1}{\lambda_j} \right), \tag{16}$$

$$\epsilon_{2,j} = \lambda_j^2 \left[ \sum_{k=k_{\min}}^{T-1} |\alpha_{k|j}|^2 \left( \frac{tT}{2\pi k} \right)^2 - \frac{1}{\lambda_j^2} \right], \tag{17}$$

are useful to study the convergence of the distance $d(|x\rangle, |x_0\rangle)$ for large dimension of the clock Hilbert space, $T = 2^{n_c}$:

$$d(|x\rangle, |x_0\rangle) = \sqrt{\frac{\sum_{j,l=0}^{2^n-1} \frac{|\beta_j|^2\,|\beta_l|^2}{\lambda_j^2\,\lambda_l^2} (\epsilon_{2,j} - \epsilon_{1,j} - \epsilon_{1,l} - \epsilon_{1,j}\,\epsilon_{1,l})}{\sum_{j,l=0}^{2^n-1} \frac{|\beta_j|^2\,|\beta_l|^2}{\lambda_j^2\,\lambda_l^2} (1 + \epsilon_{2,j})}}. \tag{18}$$

The algorithm provides reliable solutions of linear systems only if $d(|x\rangle, |x_0\rangle)$ approaches zero, namely for small $\epsilon_{1,j}$ and $\epsilon_{2,j}$.

We have computed numerically $\epsilon_{l,j}^{\mathrm{HHL}}$ ($l \in \{1, 2\}$) for the original HHL algorithm for 50 equally spaced values of $\lambda_j \in (0, 1)$, 50 equally spaced values of $t \in (0.1\,\pi, \pi)$, and $n_c$ ranging from 3 to 9 ($T \in [2^3, 2^9]$). We also assume $k_{\min} = 1$, that is the worst choice as higher values allow for shorter times of the amplitude amplification $O(1/\sqrt{p_0})$ which decreases with increasing $C = 2\pi k_{\min}/t_0$. The numerical data are then fitted with the function $a_l(\lambda_j tT)^{-2}$. The fits, shown in Fig. 2, result in $a_1 \simeq 9.94$ and $a_2 \simeq 31.54$ with a very good agreement. Given $\lambda_j \geqslant \lambda_{\min} \geqslant 1/\kappa$, the error of the orignal HHL algorithm is

$$d\left(|x\rangle, \left|x_0^{\mathrm{HHL}}\right\rangle\right) \propto \frac{\kappa}{T}. \tag{19}$$

For the HHL variant, $\epsilon_{1,j}^{\mathrm{var}}$ decreases with $T$ slower than $\epsilon_{1,j}^{\mathrm{HHL}}$. In Appendix B we compute the following bound:

$$\left|\epsilon_{1,j}^{\mathrm{var}}\right| \leqslant O\left( \frac{\ln(\lambda_j\, t\, T)}{\lambda_j\, t\, T} \right), \tag{20}$$

whose scaling with $\lambda_j tT$ is saturated by the exact numerical computations. On the other hand, we prove in Appendix C that the convergence of $\epsilon_{2,j}^{\mathrm{var}}$ for the HHL variant depends on $k_{\min}$ (and therefore $\kappa'$):

$$\sin^2\left(\frac{\lambda_j tT}{2}\right)\left(\frac{2\kappa'}{tT}\right)^2 + O\left(\frac{\ln(\lambda_j tT)}{\lambda_j tT}\right) \leqslant \left|\epsilon_{2,j}^{\mathrm{var}}\right| \leqslant O\left(\frac{\kappa'}{tT}\right) + O\left(\frac{\ln(\lambda_j tT)}{\lambda_j tT}\right). \tag{21}$$

Recalling $\lambda_j \geqslant \lambda_{\min} \geqslant 1/\kappa$, the lower bound of the computational error for the HHL variant is then

$$d\left(|x\rangle, \left|x_0^{\mathrm{var}}\right\rangle\right) \geqslant \begin{cases} O\left(\sqrt{\frac{\kappa}{T}\,\ln\frac{T}{\kappa}}\right) & \text{if } \kappa' = O\left(\sqrt{\kappa\, T\, \ln\left(\frac{T}{\kappa}\right)}\right) \\ \sin^2\left(\frac{\lambda_j tT}{2}\right)\left(\frac{2\kappa'}{tT}\right)^2 & \text{if } \kappa' \gg O\left(\sqrt{\kappa\, T\, \ln\left(\frac{T}{\kappa}\right)}\right) \end{cases}. \tag{22}$$

The inequalities (21) and (22) imply that the HHL variant is accurate when both the condition number $\kappa$ and its computable bound $\kappa'$ are much smaller than $T$, resulting in $k_{\min} = \lfloor tT/(4\pi\kappa')\rfloor \gg 1$. The HHL variant is then not accurate if the bound $\kappa'$ is as large as $T$ ($k_{\min} \approx 1$) even when $\kappa \ll T$.

The convergence problem of the HHL variant can be understood from Eq. (12). Recalling that $C^{\rm var} = H^{\otimes n_c}$ in this case, the coefficients of the states with fixed $|\gamma\rangle_c$ are proportional to

$$\frac{\alpha^{\rm var}_{k|j}}{T}\sum_{\sigma=0}^{T-1} e^{i\frac{\sigma}{T}(2\pi k-\lambda_j t_0)+i\pi\sum_{l=1}^{n_c}\sigma_l\gamma_l} \tag{23}$$

which, for $\gamma = 0$ and from Eq. (10), is equal to $|\alpha_{k|j}|^2$, and is centered at $k \approx \lambda_j t_0/(2\pi)$. Nevertheless, when $\gamma \neq 0$, the phase is shifted and the final distribution is not centered around the correct value. This problem is avoided by measuring the clock in the computational basis and postselecting the outcome $\gamma = 0$, as discussed in the next section.

## IV. IMPROVED HHL VARIANT

We have shown that the HHL variant becomes unreliable when the condition number is poorly estimated. An improved HHL variant, that is accurate even without any bound $\kappa'$, can be implemented with the amplitude amplification [24,27] of the component $|1\rangle_a|0\rangle_c$, instead of $|1\rangle_a$, after the circuit that leads to the state (8). The probability $\tilde{p}$ of measuring $|1\rangle_a|0\rangle_c$ from the state (8) is

$$\tilde{p} = \sum_{j=0}^{2^n-1}|\beta_j|^2\left(\sum_{k=k_{\min}}^{T-1}\left|\alpha^{\rm var}_{k|j}\right|^2\frac{CtT}{2\pi k}\right)^2. \tag{24}$$

The amplitude amplification will require an average number of calls to the circuit in Fig. 1 of order $\sqrt{1/\tilde{p}}$. We are more restrictive on the state component to be amplified, $\tilde{p} < p_0$, but the bound (20) and $C = 2\pi k_{\min}/t_0 \simeq 1/(2\kappa')$ prove that $\tilde{p}$ and $p_0$ are both of order $O(\lambda_{\min}\kappa')^{-2} = O(\kappa/\kappa')^2$. Therefore, the HHL variant and the improved algorithm have the same scaling of the running time.

After the amplitude amplification of the state $|1\rangle_a|0\rangle_c$, the output state is

$$|\tilde{x}\rangle = \frac{1}{\sqrt{\tilde{p}}}\sum_{j=0}^{2^n-1}\sum_{k=k_{\min}}^{T-1}\beta_j\left|\alpha^{\rm var}_{k|j}\right|^2\frac{CtT}{2\pi k}|1\rangle_a|0\rangle_c|u_j\rangle_r. \tag{25}$$

In analogy to the state (13) that encodes the exact solution $\mathbf{x}$ in the Hilbert space of quantum registers, the state (25) encodes the approximate solution $\tilde{\mathbf{x}} = \sum_{j=0}^{2^n-1}\beta_j\sum_{k=k_{\min}}^{T-1}|\alpha^{\rm var}_{k|j}|^2\frac{tT}{2\pi k}\mathbf{u}_j$. We then measure the error of the estimate through the fidelity between $|\tilde{x}\rangle$ and $|x\rangle$:

$$\begin{aligned}\langle x|\tilde{x}\rangle &= \frac{C^2}{\sqrt{p\tilde{p}}}\sum_{j=0}^{2^n-1}\frac{|\beta_j|^2}{\lambda_j}\sum_{k=1}^{T-1}\left|\alpha^{\rm var}_{k|j}\right|^2\frac{tT}{2\pi k}\\ &= \frac{\sum_{j=0}^{2^n-1}\frac{|\beta_j|^2}{\lambda_j}\sum_{k=1}^{T-1}\left|\alpha^{\rm var}_{k|j}\right|^2\frac{tT}{2\pi k}}{\sqrt{\sum_{j=0}^{2^n-1}\frac{|\beta_j|^2}{\lambda_j^2}\sum_{l=0}^{2^n-1}|\beta_l|^2\left(\sum_{k=1}^{T-1}\left|\alpha^{\rm var}_{k|l}\right|^2\frac{tT}{2\pi k}\right)^2}}.\end{aligned} \tag{26}$$

The difference between Eqs. (26) and (15) is that the same sum appears in both the numerator and the denominator of (26). Then, the convergence of the distance $d(|x\rangle,|\tilde{x}\rangle)$ depends only on the quantity $\epsilon^{\rm var}_{1,j}$ (16) and on its bound (20). Using the notation $1/\tilde{\lambda}_j := \sum_{k=k_{\min}}^{T-1}|\alpha^{\rm var}_{k|j}|^2 tT/(2\pi k)$, the computational error is given by

$$d(|x\rangle,|\tilde{x}\rangle) = \sqrt{\frac{\sum_j\frac{|\beta_j|^2}{\lambda_j^2}\sum_l\frac{|\beta_l|^2}{\lambda_l\tilde{\lambda}_l}\left(\frac{\lambda_l}{\tilde{\lambda}_l}-\frac{\lambda_j}{\tilde{\lambda}_j}\right)}{\sum_j\frac{|\beta_j|^2}{\lambda_j^2}\sum_l\frac{|\beta_l|^2}{\tilde{\lambda}_l^2}}}. \tag{27}$$

Since Eq. (20) implies that $\lambda_j/\tilde{\lambda}_j = 1 + O[\ln(\lambda_j tT)/(\lambda_j tT)]$, the error is bigger if the eigenvalue is smaller. Then, recalling $\lambda_{\min} = \min_j\lambda_j$, the error is bounded by

$$d(|x\rangle,|\tilde{x}\rangle) \leqslant O\left(\sqrt{\frac{\ln(\lambda_{\min}tT)}{\lambda_{\min}tT}}\right) \leqslant O\left(\sqrt{\frac{\kappa}{tT}\ln\frac{tT}{\kappa}}\right). \tag{28}$$

Equation (28) shows that the error can be arbitrary small by choosing a big enough number of qubits $n_c$ in the $c$ register, with $T = 2^{n_c}$, provided that the condition number $\kappa$ is not as large as $T$.

The quantum states $|\tilde{x}\rangle$ and $|x\rangle$ have unit normalization; therefore, vectors $\tilde{\mathbf{x}} \in \mathbb{C}^d$ that differ only in the normalization have the same encoding $|\tilde{x}\rangle$. In this sense, the states $|x\rangle$ and $|\tilde{x}\rangle$ encode only the direction of the vectors $\mathbf{x}$ and $\tilde{\mathbf{x}}$ respectively. Nevertheless, the norms of the vectors $\tilde{\mathbf{x}}$ and $\mathbf{x}$ defined above are

$$||\tilde{\mathbf{x}}||_2 = \frac{\sqrt{\tilde{p}}}{C}, \quad ||\mathbf{x}||_2 = \frac{\sqrt{p}}{C}. \tag{29}$$

The probability $\tilde{p}$ can be measured implementing the amplitude estimation algorithm (a combination of the amplitude amplification and the quantum phase estimation algorithms) [24,27] that uses an average number of calls to the circuit of Fig. 1 of order $1/\sqrt{\tilde{p}}$. In order to prove that $||\tilde{\mathbf{x}}||_2$ well approximates $||\mathbf{x}||_2$, use Eq. (20) and $\lambda_{\min} = \min_j\lambda_j$ to compute

$$\frac{\left|\,||\mathbf{x}||_2^2 - ||\tilde{\mathbf{x}}||_2^2\,\right|}{||\mathbf{x}||_2^2} \leqslant \frac{\sum_j\frac{|\beta_j|^2}{\lambda_j^2}\left|1-\frac{\lambda_j^2}{\tilde{\lambda}_j^2}\right|}{\sum_j\frac{|\beta_j|^2}{\lambda_j^2}} = O\left(\frac{\ln(\lambda_{\min}tT)}{\lambda_{\min}tT}\right). \tag{30}$$

The bound (30) proves that the norm of the solution can be measured with arbitrary small error and with the same running time scaling required for the amplification of the amplitude of state $|\tilde{x}\rangle$.

The components of $\mathbf{x}$ cannot be computed efficiently using this algorithm, as discussed in Sec. II. Nevertheless, the algorithm allows us to efficiently compute expectation values $\mathbf{x}^T M \mathbf{x}$ if $M$ is a matrix that can be efficiently implemented on a quantum computer. In order to prove it, assume without loss of generality that $M$ is Hermitian with eigenvalues $\{m_j\}_{j=1,\ldots,d}$ and eigenvectors $\{\mathbf{m}_j\}_{j=1,\ldots,d}$, otherwise apply the following argument to the Hermitian and the anti-Hermitian part of $M$ independently. Perform then a projective measurement on the output of the algorithm, $|\tilde{x}\rangle$, with projectors $\{|m_j\rangle\langle m_j|\}_{j=0,\ldots,T-1}$. After $\mu$ of such measurements, the statistical frequency of obtaining the outcome $m_j$ approximates the probability $|\langle\tilde{x}|m_j\rangle|^2$ up to an error $\delta$ with probability $1-O(e^{-2\mu\delta^2})$ from the Chernoff-Hoeffding bound [74,75]. When the sampling error $\delta$ is comparable with the computational errors (28), one requires $\mu > O[\lambda_{\min} t\, T/\ln(\lambda_{\min} t\, T)]$ repetitions for a good success probability in the Chernoff-Hoeffding bound. These repetitions can however be parallelized, without then increasing the computational time.

From the frequency of obtaining the state $|m_j\rangle$ and from the meaurement of the norm discussed above, one computes

$$
\begin{aligned}
\|\tilde{\mathbf{x}}\|_2^2 \sum_{j=0}^{T-1} m_j |\langle\tilde{x}|m_j\rangle|^2 &= \|\tilde{\mathbf{x}}\|_2^2\, \langle\tilde{x}|M|\tilde{x}\rangle = \tilde{\mathbf{x}}^T M \tilde{\mathbf{x}} \\
&= \mathbf{x}^T M \mathbf{x}\left[1 + O\left(\frac{\ln(\lambda_{\min} t\, T)}{\lambda_{\min} t\, T}\right)\right],
\end{aligned} \tag{31}
$$

achieving a good estimate of $\mathbf{x}^T M \mathbf{x}$ for large $T$.

## V. COMPLEXITY WITH DIGITAL HAMILTONIAN SIMULATION

We analyze and compare here the complexity of the above algorithms. While more recent quantum linear system solvers [11] achieve optimal asymptotic scaling, their implementation typically requires significantly more complex primitives. The comparison presented here is therefore intended to assess the relative performance of HHL-type algorithms within their natural regime of applicability. The relevant parameters for this analysis are the computational error $\epsilon$, the condition number $\kappa = \max_j |\lambda_j| / \min_j |\lambda_j|$ of the matrix $A$, and its sparsity $s$ that is the maximum number of nonvanishing elements in each row or column.

The simulation of the controlled gates (6) within error $\epsilon'$ in the quantum phase estimation can be done with a number $O(s\, T \max_{i,j} |A_{i,j}| + \frac{\ln(1/\epsilon')}{\ln\ln(1/\epsilon')})$ of queries to the matrix elements of $A$ [46,47] and additional $O((\ln d)\, s\, T \max_{i,j} |A_{i,j}| + \frac{(\ln d)\ln(1/\epsilon')}{\ln\ln(1/\epsilon')})$ one- and two-qubit gates. Note that the assumption $0 < \lambda_j \leqslant 1$ implies $\max_{i,j} |A_{i,j}| \leqslant 1$ [76]. The implementation of the quantum Fourier transform requires a number $O(\log_2^2 T)$ of one- and two- qubit gates [1,2] and the controlled rotations (7) require $O(T \log_2^2 T)$ of such operations (see Appendix A). Moreover, the circuit in Fig. 1 has to be executed $O(1/\sqrt{\tilde{p}}) = O(\kappa'/\kappa)$ times to implement the postselection. The number of sequential queries is called *query complexity*, $\mathcal{Q} = O(s\, T\, \kappa'/\kappa)$, and the number of consecutive additional gates is called *gate complexity*, $\mathcal{G} = O((s \log_2 d + \log_2^2 T)\, T\, \kappa'/\kappa)$. We focus on the gate complexity as the query complexity is proportional to $\mathcal{G}$ up to logarithmic corrections.

The gate complexity can be expressed in terms of the computational error quantified by the distance between the output state and the ideal state [see Eqs. (18) and (27)]. From Eq. (19), consider the computational error $\epsilon = \kappa/T$, and the gate complexity of the original HHL algorithm becomes $\mathcal{G} = O((s \log_2 d + \log_2^2(\kappa/\epsilon))\, \kappa'/\epsilon)$. Near-term experimental realizations and applications of HHL algorithms as subroutines [29–33,56] could benefit from avoiding the preparation of the initial entangled state of the clock (4). We then analyze the gate complexity of the HHL variant and of the improved HHL variant.

For the HHL variant, we consider the best computational error, namely the scaling of the lower bound (22), while the computational error of the improved HHL variant is given by the upper bound (28) which captures the exact numerical scaling. The computational error is then $\epsilon = \sqrt{\frac{\kappa}{T}}\, \ln(\frac{T}{\kappa})$ for the HHL variant with $\kappa' = O(\sqrt{T\kappa \ln T/\kappa})$ and for the improved HHL variant for any $\kappa'$. Inverting this relation, we obtain $T = -\frac{\kappa}{\epsilon^2} W_{-1}(-\epsilon^2)$ where $W_{-1}(z)$ is the branch of the Lambert function defined as the real solution of $We^W = z$ with largest absolute value $|W|$ [77].

The gate complexity in this case becomes

$$
\begin{aligned}
\mathcal{G} = O\Bigg(&\left(s \log_2 d + \log_2^2\left(-\frac{\kappa'}{\epsilon^2} W_{-1}(-\epsilon^2)\right)\right) \\
&\times \frac{\kappa'}{\epsilon^2}[-W_{-1}(-\epsilon^2)]\Bigg).
\end{aligned} \tag{32}
$$

Using the notation $\tilde{O}(\cdot)$ which denotes orders $O(\cdot)$ up to multiplicative powers of the logarithm of its argument, the gate complexity (32) reads $\mathcal{G} = \tilde{O}(\kappa'\, s\, \epsilon^{-2} \log_2 d)$ because $W_{-1}(-\epsilon^2) \sim 2\ln\epsilon$ for $\epsilon \to 0$. The condition $\kappa' = O(\sqrt{T\kappa \ln T/\kappa})$ for the HHL variant is compatible with $\kappa \leqslant \kappa'$ only if $\kappa \leqslant \chi\sqrt{T\kappa \ln T/\kappa}$ for some constant $\chi$. This inequality implies $\kappa \leqslant T\, e^{-W_0(1/\chi^2)} = T\, \chi^2\, W_0(1/\chi^2)$, where $W_0(z)$ is the principle branch of the Lambert function defined as the unique real solution of $We^W = z$ for positive $z$ [for instance $W_0(1) \simeq 0.57$] [77].

Equation (22) also implies that the computational error of the HHL variant with $\kappa'$ as large as $T$ ($\kappa' \sim T$) is a non-negligible constant, and the algorithm output is not reliable. If, instead, $\kappa' \gg \sqrt{T\kappa \ln T/\kappa}$ then $\kappa \ll \tilde{O}[(\kappa')^2/T]$ and the gate complexity is lower bounded by $\mathcal{G} = \tilde{O}(T\kappa'\kappa^{-1} s \log_2 d) \gg \tilde{O}(T^2(\kappa')^{-1} s \log_2 d)$. This lower bound equals the gate complexity $\tilde{O}(\kappa'\, s\, \epsilon^{-2} \log_2 d)$ in (32) that holds only for the improved variant in this regime, given the computational error $\epsilon \sim \kappa'/T$ as in (22). In conclusion, the improved algorithm is more efficient than (if $\kappa' \gg \sqrt{T\kappa \ln T/\kappa}$) or at worst as efficient as [if $\kappa' = O(\sqrt{T\kappa \ln T/\kappa})$] the HHL variant in terms of computational error, query, and gate complexities. Furthermore, the improved algorithm is independent of having an estimate of the condition number, if we assume $k_{\min} = 1$ such that $1/\kappa' = 4\pi/t_0 = 4\pi/(tT)$ [see the discussion after Eq. (10)] scales as the machine precision, and thus eliminates the preprocessing time $O(sd)$ required to obtain this estimate.

These complexity scalings assume queries to matrix elements of $A$. To translate them into concrete time and space complexities, an explicit physical implementation of such queries is therefore required. For general matrices $A$, known constructions require $\mathcal{M} = O(d^2)$ memory qubits and $\mathcal{G}_{\mathrm{O}} = O(\ln d)$ gates [67,78]. For general $s$-sparse matrices (assuming that the positions of nonzero matrix elements are known), the memory requirement reduces to $O(s\,d)$. If the matrix $A$ is a sum of $P$ unitary matrices in $SU(2)^{\otimes n}$, the building oracles for the Hamiltonian simulation require $\mathcal{M} = O(nP)$ memory qubits and $\mathcal{G}_{\mathrm{O}} = O(T\ln(nP))$ gates [79,80]. An important subclass of such matrices is given by $k$-local Hamiltonians, in which each term acts nontrivially on at most $k$ of the $n$ qubits. The overall time complexity of the HHL algorithm and its variants is $\mathcal{G}_{\mathrm{O}}\,\mathcal{Q} + \mathcal{G}$, while the space complexity is $\mathcal{M} + \log_2 d + \log_2 T + 1$.

## VI. COMPLEXITY WITH ANALOGICAL QUANTUM SIMULATORS

An alternative computational model exploits analog quantum simulators [48–51], instead of digital Hamiltonian simulation discussed in the previous section. In order to integrate an analog quantum simulator into HHL-type algorithms, first notice that the controlled unitary (6) is the product of $n_c$ unitaries $e^{iA\,t_0\,2^{j-1}\tau_j/T}$, each one controlled by the $j$ qubit of the clock for $j = 1, \dots, n_c$ [see the rightmost side of (6)]. A unitary controlled by a single qubit can be implemented using only the unitaries $e^{iA\,t_0\,2^{j-1}\tau_j/T}$ without control together with a constant overhead of elementary gates [81–83]. If the register $r$ is engineered such that its Hamitonian is proportional to $A$ using analog quantum simulators, then the full transformation (6) can be implemented by letting the register $r$ freely evolve at certain times for durations proportional to $\tau_j$, and using $O(n_c) = O(\log T)$ additional elementary gates.

Natural matrices that can be implemented on analog quantum simulators are 2-local Hamiltonians. Although the Hamiltonian contains $O(n^2)$ coupling terms, these couplings are controlled through global or parallel parameters [48–51]. As a consequence, their simultaneous tuning does not introduce an additional time overhead scaling with the system size, unlike what would occur in a digital circuit model. The simulations of uncontrolled unitaries $e^{iA\,t_0\,\tau/T}$ then require no additional memory qubits and runtime $O(T)$. Therefore, the time complexity equals the gate complexity $\mathcal{G}$ and the space complexity is $\log_2 d + \log_2 T + 1$. For the HHL variant with $\kappa' = O(\sqrt{T\kappa \ln T/\kappa})$ and for the improved HHL variant with any $\kappa'$, one finds (similarly to the discussion in Sec. V) $\mathcal{G} = O((\log_2^2 T)\,T\,\kappa'/\kappa) = \tilde{O}(\kappa'/\epsilon^2)$ that equals Eq. (32) without the term $s\log d$. By contrast, if $\kappa' \gg \sqrt{T\kappa \ln T/\kappa}$ and $\epsilon \sim \kappa'/T$ for the HHL variant [see Eq. (22)], the gate complexity is much larger, $\mathcal{G} = \tilde{O}(T\kappa'/\kappa) \gg \tilde{O}(\kappa'/\epsilon^2)$.

For the special class of two-local Hamiltonians $A$, the independence of the gate complexity from $d$ (or $n$) and the absence of quantum memory overhead $\mathcal{M}$ imply that the complexity of HHL-type algorithms, implemented with analog quantum simulators, exhibits more favorable scalings with $d$ than more recent algorithms based on block-encoding or other oracle-based Hamiltonian simulations [11], which do require these overheads. The existence and the identification of these special instances of linear problems are therefore relevant, since there also exists problem instances solved by block-encoding based quantum algorithms [79,80] that share the same complexity scaling with respect to the problem size $d$ of HHL algorithms, but exhibit more favorable dependence with the target error $\epsilon$.

## VII. NUMERICAL EXAMPLES

In this section, we simulate the HHL variant and the improved algorithm for the solution of two dimensional linear problems ($d = 2$) using the PYTHON library QISKIT, without implementing environmental noise on the quantum gates. The choice $d = 2$ allows us to consider larger numbers of qubits in the clock register $n_c \in [3,11]$ without changing the error scaling with $n_c$. We then need only one qubit in the $r$ register, and fix the parameters, $k_{\min} = 1$, $t = 8\pi/5$, and $C = 2\pi/t_0$. The condition $k_{\min} = 1$ corresponds to the worst case scenario without any bound of the condition number. For these numbers $n_c$, our classical simulation stores efficiently the full state vector, and then the amplitude amplification is replaced by the readout of the corresponding state coefficient. The code can be find on GitHub [84].

We randomly generated 50 linear problems of the form (1) defined by Hermitian matrices $A$ with eigenvalues in the range (0,1] and vectors $b$ with unit norm. To generate such random matrices $A$, we sample diagonal matrices $A' = \left(\begin{smallmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{smallmatrix}\right)$ from the uniform distribution of $\lambda_j \in (0, 1]$ and transform them through a random unitary matrix, $A = UA'U^\dagger$, with $U$ drawn from the Haar measure [85]. In Fig. 3, we plot the infidelity (left panel) and the norm error (right panel) as function of the number $n_c$ of qubits of the clock. We compared the HHL variant with postselection of the state $|1\rangle_a$ (green dots) and our improved algorithm with postselection of the state $|1\rangle_a|0\rangle_c$ (blue crosses), and the lines represent the average behaviors. The plots show that the errors of the HHL variant remain constant, while the errors of the new algorithm decrease to zero with increasing $n_c$, in agreement with the analytical computations in Secs. III and IV.

The errors of the HHL algorithm and of its variants for general $d$ are given by the analytical bounds in Secs. III and IV.

## VIII. NEGATIVE EIGENVALUES OF *A*

In this section, we discuss how to apply the proposed algorithm to the case of nonpositive eigenvalues of $A$. Using the normalization argument described in Sec. II, a general linear problem (1) is transformed into a linear problem with $-1 \leqslant \lambda_j \leqslant 1$.

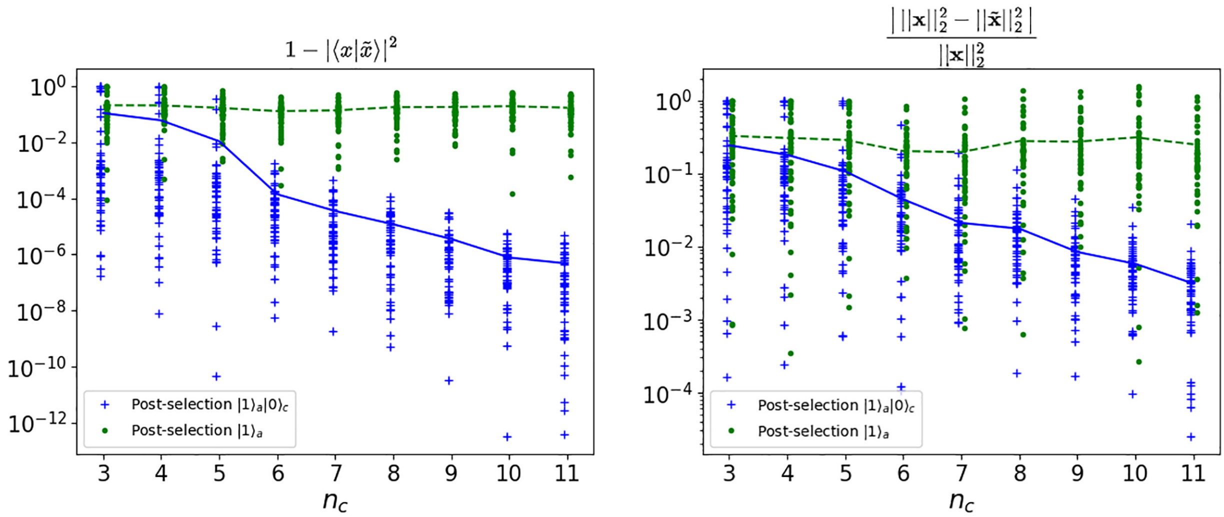


FIG. 3. Log-lin plot of the infidelity (left panel) and of the norm error (right panel) between the output state and the ideal state of 50 randomly generated linear problems. Green dots corresponds to the HHL variant with postselection of the state $|1\rangle_a$, whose average behaviors are the dashed green lines. Blue crosses are the errors of the improved algorithm, and the solid blue lines are the averages.

First, set $t_0 = tT$ as before but with $0 < t < \pi$. After the phase estimation algorithm the distribution $|\alpha_{k|j}|^2$ is centered around $k \sim \lambda_j tT/(2\pi) \bmod T$. When the eigenvalue is positive $\lambda_j > 0$, the most probable value is $k \sim \lambda_j tT/(2\pi) < T/2$. When the eigenvalue is negative $\lambda_j < 0$, the most probable value is $k \sim \lambda_j tT/(2\pi) \bmod T > T/2$. In other words, the periodicity of the $\alpha_{k|j}$ is exploited to differentiate between positive and negative eigenvalues. Next, the angles $\theta(k)$ of the conditional rotation (7) are defined by

$$\sin[\theta(k)] := \begin{cases} \dfrac{C\,t_0}{2\pi k} & \text{if } k \leqslant T/2 \\ \dfrac{C\,t_0}{2\pi(k-T)} & \text{if } k \geqslant T/2 \end{cases}. \tag{33}$$

The bounds (20) and (21) hold true also for this case and so do all the conclusions. The difference in the computations arises in Eq. (B1), where we should use the new Eq. (33). We also have to consider the cases $0 < t\,\lambda_j/(2\pi) < 1/2$ and $-1/2 < t\,\lambda_j/(2\pi) < 0$ separately in Appendix B. There are divergences for $|t\,\lambda/(2\pi)| \approx 1/2$ which are avoided by choosing $t \neq \pi$. With this little modification, more general matrices can be handled.

## IX. CONCLUSIONS

We analyzed the HHL algorithm for solving linear systems (1) on quantum computers, as well as a variant algorithm that is often considered for practical implementations [29–33]. We focused on the scaling of the performances of these algorithms with the number of qubits $n_c$ of the clock, which is a relevant parameter as it gives the precision of the quantum phase estimation for eigenvalue computations and the number of required gates. Another relevant parameter is the condition number that is usually assumed to have a known upper bound, and we extended the error analysis to the case where such bound is unknown. We proved that, in this case, the error of the HHL variant, unlike that of the original HHL algorithm, does not converge for large $n_c$ but instead oscillates around a constant value. This nonconvergent behavior arises from oscillatory terms in the amplified amplitude.

We proposed a modified algorithm, based on the amplification of a smaller amplitude without detrimental oscillating terms. This modification preserves the simplicity of the original variant while restoring full convergence of the output error as the clock-register size increases. We confirmed this result with numerical simulations. Furthermore, the modified algorithm does not assume a bound of the condition number whose computation requires running time $O(sd)$ and dominates the running time scaling with $d$ of the quantum algorithm.

We also considered HHL algorithms with unitary simulations implemented with analog quantum simulators. This approach naturally identifies a class of linear problems for which such algorithms are competitive with more recent quantum linear solvers for moderate errors. For these instances, the complexity of HHL algorithms exhibits more favorable scaling with the problem size, at the expense of a less advantageous dependence on the computational error.

These findings are relevant for both theoretical developments and near-term implementations of quantum linear solvers as subroutines in larger quantum algorithms, particularly on NISQ platforms. In particular, practical implementations with currently available hardware may not have

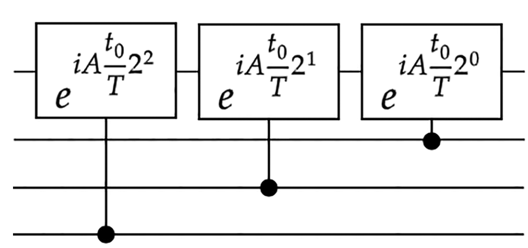


FIG. 4. Implementation of the conditional evolution operation for a clock with three qubits.

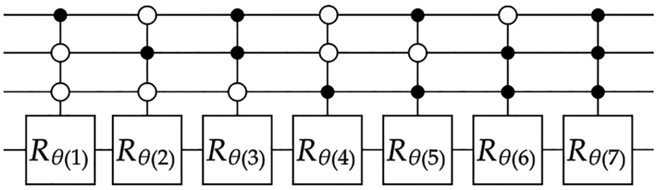


FIG. 5. Implementation of the conditional rotation described in Eq. (7) with $k_{\min} = 1$ and $n_c = 3$ qubits of the clock.

sufficient gate depth and memory to implement more recent algorithms as those discussed in [11].


### ACKNOWLEDGMENT

We thank Seth Lloyd and Giacomo De Palma for useful discussions.


### DATA AVAILABILITY

The data that support the findings of this article are openly available [84].

### APPENDIX A: STEPS OF THE HHL VARIANT

In this appendix, we explicitly show intermediate states of the circuit in Fig. 1. The first operator $\mathcal{C}$ is the initialization of the clock register that differentiates the original HHL algorithm from its variant, as discussed in Sec. II. The second gate (6) can be realized using the circuit represented in Fig. 4 [22] where each of the exponentials can be implemented if the matrix $A$ is sparse or if it is a linear combination of unitaries [38–47]. The state after these operations is

$$\left|\psi_1^{\mathrm{HHL}}\right\rangle = |0\rangle_a \sqrt{\frac{2}{T}} \sum_{j=0}^{2^n-1} \sum_{\tau=0}^{T-1} \beta_j \sin\left(\frac{\pi(2\tau+1)}{2T}\right) \times e^{i\frac{\tau}{T}\lambda_j t_0} |\tau\rangle_c |u_j\rangle_r, \tag{A1}$$

$$\left|\psi_1^{\mathrm{var}}\right\rangle = |0\rangle_a \frac{1}{\sqrt{T}} \sum_{j=0}^{2^n-1} \sum_{\tau=0}^{T-1} \beta_j e^{i\frac{\tau}{T}\lambda_j t_0} |\tau\rangle_c |u_j\rangle_r, \tag{A2}$$

for the orginal HHL algorithm and for the HHL variant respectively.

After the inverse quantum Fourier transform the state vector can be written in the following compact form for both the HHL algorithm and the HHL variant,

$$|\psi_2\rangle = |0\rangle_a \sum_{j=0}^{2^n-1} \sum_{k=0}^{T-1} \beta_j\, \alpha_{k|j}\, |k\rangle_c |u_j\rangle_r, \tag{A3}$$

where the coefficients $\alpha_{k|j}$ are defined in Eqs. (9) and (10). Since the state $|\psi_2\rangle$ has been achieved through unitary operations, it is normalized such that

$$1 = \langle\psi_2|\psi_2\rangle = \sum_{k=0}^{T-1} |\alpha_{k|j}|^2. \tag{A4}$$

The next gate, namely the controlled unitary defined by the transformation (7), can be implemented using $2^{n_c} - k_{\min}$ multiqubit controlled rotations, as exemplified in Fig. 5. Each multiqubit controlled rotation requires $O(n_c^2)$ elementary gates [86], and thus the transformation (7) is implemented in time $O(n_c^2\, 2^{n_c})$. The state vector after this step is

$$\begin{aligned} |\psi_3\rangle = & \sum_{j=0}^{2^n-1} \sum_{k=k_{\min}}^{T-1} \beta_j\, \alpha_{k|j}\, \sin[\theta(k)] |1\rangle_a |k\rangle_c |u_j\rangle_r \\ & + \sum_{j=0}^{2^n-1} \sum_{k=k_{\min}}^{T-1} \beta_j\, \alpha_{k|j}\, \cos[\theta(k)] |0\rangle_a |k\rangle_c |u_j\rangle_r \\ & + \sum_{j=0}^{2^n-1} \sum_{k=0}^{k_{\min}-1} \beta_j\, \alpha_{k|j}\, |0\rangle_a |0\rangle_c |u_j\rangle_r, \end{aligned} \tag{A5}$$

recalling $\sin\theta(k) := Ct_0/(2\pi k)$.

The quantum Fourier transform then yields the state vector

$$\begin{aligned} |\psi_4\rangle = & \frac{1}{\sqrt{T}} \sum_{j=0}^{2^n-1} \sum_{\sigma=0}^{T-1} \sum_{k=k_{\min}}^{T-1} \beta_j\, \alpha_{k|j}\, \sin[\theta(k)]\, e^{i\frac{\sigma}{T}2\pi k} |1\rangle_a \\ & \times |\sigma\rangle_c |u_j\rangle_r + \frac{1}{\sqrt{T}} \sum_{j=0}^{2^n-1} \sum_{\sigma=0}^{T-1} \sum_{k=k_{\min}}^{T-1} \beta_j\, \alpha_{k|j}\, \cos[\theta(k)] \\ & \times e^{i\frac{\sigma}{T}2\pi k} |0\rangle_a |\sigma\rangle_c |u_j\rangle_r + \frac{1}{\sqrt{T}} \sum_{j=0}^{2^n-1} \sum_{\sigma=0}^{T-1} \sum_{k=0}^{k_{\min}-1} \\ & \times \beta_j\, \alpha_{k|j}\, |0\rangle_a |\sigma\rangle_c |u_j\rangle_r. \end{aligned} \tag{A6}$$

After the inverse of the transformation (6) and the last gate $\mathcal{C}^{-1}$, one obtains the state (8).

### APPENDIX B: COMPUTATION OF THE BOUND (20)

In this appendix we derive the bound (20). First, use the identity (A4) and the triangle inequality to compute

$$\begin{aligned} |\epsilon_{1,j}| = \lambda_j \left| \frac{1}{\lambda_j} - \sum_{k=k_{\min}}^{T-1} |\alpha_{k|j}^{\mathrm{var}}|^2 \frac{tT}{2\pi k} \right| &= \left| \sum_{k=0}^{k_{\min}-1} |\alpha_{k|j}^{\mathrm{var}}|^2 + \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{T^2} \sum_{k=k_{\min}}^{T-1} \frac{\left(k - \frac{\lambda_j t T}{2\pi}\right)}{k \sin^2\left[\frac{\pi}{T}\left(k - \frac{\lambda_j t T}{2\pi}\right)\right]} \right| \\ &\leqslant \sum_{k=0}^{k_{\min}-1} |\alpha_{k|j}^{\mathrm{var}}|^2 + \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{T^2} \sum_{k=k_{\min}}^{T-1} \frac{\left|k - \frac{\lambda_j t T}{2\pi}\right|}{k \sin^2\left[\frac{\pi}{T}\left(k - \frac{\lambda_j t T}{2\pi}\right)\right]}. \end{aligned} \tag{B1}$$

In order to bound the sums in the right-hand-side of (B1), we will extensively use the following inequalities:

$$\sin(\pi x) \geqslant \begin{cases} 2x & \text{if } 0 \leqslant x \leqslant \frac{1}{2} \\ 2(1-x) & \text{if } \frac{1}{2} \leqslant x \leqslant 1 \end{cases}, \tag{B2}$$

$$\sum_{x=a}^{b} f(x) \leqslant \int_a^b f(x)dx + f(a) + f(b), \tag{B3}$$

valid for positive convex functions $f(x)$. The two cases in inequality (B2) require us to separatedly treat different contributions in the upper bound (B1).

Let us start with considering $t\lambda_j/(2\pi) \leqslant 1/2$. The first term in (B1) is bounded as follows:

$$\begin{aligned}\sum_{k=0}^{k_{\min}-1} |\alpha_{k|j}^{\text{var}}|^2 &= \sum_{k=0}^{k_{\min}-1} \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2 \sin^2\left[\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k\right)\right]} \leqslant \frac{1}{4}\sin^2\left(\frac{\lambda_j tT}{2}\right) \sum_{k=0}^{k_{\min}-1}\left(\frac{\lambda_j tT}{2\pi} - k\right)^{-2} \\ &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\left[\int_{k=0}^{k_{\min}-1}\left(\frac{\lambda_j tT}{2\pi} - k\right)^{-2} dk + \left(\frac{2\pi}{\lambda_j tT}\right)^2 + \left(\frac{\lambda_j tT}{2\pi} - k_{\min} + 1\right)^{-2}\right] \\ &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\left[\frac{k_{\min} - 1}{\frac{\lambda_j tT}{2\pi}\left(\frac{\lambda_j tT}{2\pi} - k_{\min} + 1\right)} + \left(\frac{2\pi}{\lambda_j tT}\right)^2 + \left(\frac{\lambda_j tT}{2\pi} - k_{\min} + 1\right)^{-2}\right] \\ &\leqslant \frac{\pi}{2\lambda_j tT} + \left(\frac{2\pi}{\lambda_j tT}\right)^2 = O\left(\frac{1}{\lambda_j tT}\right),\end{aligned} \tag{B4}$$

where we have used the assumption $1 \leqslant k_{\min} < \lambda_j tT/(4\pi)$ (see Sec. II) in the last inequality.

Defining $a_j = \lfloor \lambda_j tT/2\pi \rfloor$ and $0 \leqslant \Delta_j < 1$ such that

$$\frac{\lambda_j tT}{2\pi} = a_j + \Delta_j, \tag{B5}$$

the second sum in the upper bound (B1) can be split into the following three sums:

$$\sum_{k=k_{\min}}^{T-1} = \sum_{k=k_{\min}}^{a_j} + \sum_{k=a_j+1}^{a_j+\frac{T}{2}} + \sum_{k=a_j+\frac{T}{2}+1}^{T-1}. \tag{B6}$$

Using again the inequalities (B2) and (B3), we compute for the first of these sums

$$\begin{aligned}\frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2}\sum_{k=k_{\min}}^{a_j}\frac{\frac{\lambda_j tT}{2\pi} - k}{k\,\sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k\right)\right)} &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\sum_{k=k_{\min}}^{a_j}\frac{1}{k\left(\frac{\lambda_j tT}{2\pi} - k\right)} \\ &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\left(\int_{k_{\min}}^{a_j}\frac{1}{k\left(\frac{\lambda_j tT}{2\pi} - k\right)}dk + \frac{1}{k_{\min}\left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)} + \frac{1}{\Delta_j a_j}\right) \\ &= \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\left(\frac{2\pi}{\lambda_j tT}\ln\left(\frac{a_j\left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)}{\Delta_j k_{\min}}\right) + \frac{1}{k_{\min}\left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)} + \frac{1}{\Delta_j a_j}\right) \\ &\leqslant \frac{\pi}{\lambda_j tT}\ln\left(\frac{\lambda_j tT}{2\pi}\right) + \frac{12\pi}{4\lambda_j tT} = O\left(\frac{\ln(\lambda_j tT)}{\lambda_j tT}\right),\end{aligned} \tag{B7}$$

where we have used, in the last inequality, $\sin^2(\lambda_j tT/2) = \sin^2(\pi\Delta_j)$, $0 < -\sin^2(\pi\Delta_j)\ln(\Delta_j) < 1$, and $\frac{\sin^2(\pi\Delta_j)}{\Delta_j} < 5/2$.

A very similar calculation bounds the second sum in (B6):

$$\begin{aligned}\frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2}\sum_{k=a_j+1}^{\frac{T}{2}+a_j}\frac{k - \frac{\lambda_j tT}{2\pi}}{k\,\sin^2\left(\frac{\pi}{T}\left(k - \frac{\lambda_j tT}{2\pi}\right)\right)} &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}\sum_{k=a_j+1}^{\frac{T}{2}+a_j}\frac{1}{k\left(k - \frac{\lambda_j tT}{2\pi}\right)} \\ \leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4}&\left(\frac{4}{(T - 2\Delta_j)(T + 2a_j)} + \frac{1}{(1 - \Delta_j)(a_j + 1)} + \int_{a_j+1}^{\frac{T}{2}+a_j}\frac{1}{k\left(k - \frac{\lambda_j tT}{2\pi}\right)}dk\right)\end{aligned}$$

$$
\begin{aligned}
&= \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\left(\frac{4}{(T-2\Delta_j)(T+2a_j)} + \frac{1}{(1-\Delta_j)(a_j+1)} + \frac{2\pi}{\lambda_j t T}\ln\frac{(T-2\Delta_j)(a_j+1)}{(T+2a_j)(1-\Delta_j)}\right)\\
&\leqslant \frac{1}{4}\left(\frac{4}{T(T-2)} + \frac{10\pi}{2\lambda_j t T} - \frac{2\pi\ln 4}{\lambda_j t T} + \frac{2\pi}{\lambda_j t T} + \frac{2\pi}{\lambda_j t T}\ln\left(\frac{\lambda_j t T}{2\pi}+1\right)\right) = O\left(\frac{\ln(\lambda_j t T)}{\lambda_j t T}\right).
\end{aligned} \tag{B8}
$$

The last inequality in Eq. (B8) follows from the bounds $\frac{\sin^2(\pi\Delta_j)}{1-\Delta_j} < 5/2$, $-\sin^2(\pi\Delta_j)\ln(1-\Delta_j) < 1$, and $1/2 - 1/T < (T-2\Delta_j)/(T+2a_j) < 1$ which comes from $\lambda_j t/(2\pi) \leqslant 1/2$ and implies $|\ln(T-2\Delta_j)/(T+2a_j)| < \ln 4$ for $T \geqslant 4$.

Repeating the same procedure, the contribution from the last sum in (B6) is

$$
\begin{aligned}
&\frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{T^2}\sum_{k=\frac{T}{2}+a_j+1}^{T-1}\frac{k-\frac{\lambda_j t T}{2\pi}}{k\sin^2\left(\frac{\pi}{T}\left(k-\frac{\lambda_j t T}{2\pi}\right)\right)} \leqslant \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\sum_{k=\frac{T}{2}+a_j+1}^{T-1}\frac{k-\frac{\lambda_j t T}{2\pi}}{k\left(T+\frac{\lambda_j t T}{2\pi}-k\right)^2}\\
&\leqslant \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\left(\frac{T-1-\frac{\lambda_j t T}{2\pi}}{(T-1)\left(\frac{\lambda_j t T}{2\pi}+1\right)^2} + \frac{4(T+2-2\Delta_j)}{(T+2\Delta_j-2)^2(T+2a_j+2)} + \int_{\frac{T}{2}+a_j+1}^{T-1}\frac{k-\frac{\lambda_j t T}{2\pi}}{k\left(T+\frac{\lambda_j t T}{2\pi}-k\right)^2}dk\right)\\
&= \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\left(\frac{T-1-\frac{\lambda_j t T}{2\pi}}{(T-1)\left(\frac{\lambda_j t T}{2\pi}+1\right)^2} + \frac{4(T+2-2\Delta_j)}{(T+2\Delta_j-2)^2(T+2a_j+2)}\right.\\
&\quad\left. + \frac{T}{T+\frac{\lambda_j t T}{2\pi}}\left(\frac{1}{\frac{\lambda_j t T}{2\pi}+1} - \frac{2}{T+2\Delta_j-2}\right) + \frac{\frac{\lambda_j t T}{2\pi}}{\left(T+\frac{\lambda_j t T}{2\pi}\right)^2}\ln\frac{\left(\frac{\lambda_j t T}{2\pi}+1\right)(T+2a_j+2)}{(T-1)(T+2\Delta_j-2)}\right)\\
&\leqslant \frac{1}{4}\left(\left(\frac{2\pi}{\lambda_j t T}\right)^2 + \frac{4}{T^2} + \frac{2\pi}{\lambda_j t T} + \frac{2\pi}{\lambda_j t T}\ln\left(2\left(\frac{\lambda_j t T}{2\pi}+1\right)\frac{T+1}{(T-1)^2}\right)\right) = O\left(\frac{1}{\lambda_j t T}\right).
\end{aligned} \tag{B9}
$$

Finally, the estimates (B4), (B7), (B8), and (B9) imply (20) for $0 < t\lambda_j/(2\pi) \leqslant 1/2$.

The second sum in the right-hand-side of (B1) for the remaining case $1/2 < t\lambda_j/(2\pi) < 1$ is bounded following the same steps. Split the sum as

$$
\sum_{k=k_{\min}}^{T-1} = \sum_{k=k_{\min}}^{a_j-\frac{T}{2}} + \sum_{k=a_j-\frac{T}{2}+1}^{a_j} + \sum_{k=a_j+1}^{T-1}, \tag{B10}
$$

and use the inequalities (B2) and (B3) appropriately. The bound of the first sum in the right-hand-side of (B10) is

$$
\begin{aligned}
&\frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{T^2}\sum_{k=k_{\min}}^{a_j-\frac{T}{2}}\frac{\frac{\lambda_j t T}{2\pi}-k}{k\sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j t T}{2\pi}-k\right)\right)} \leqslant \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\sum_{k=k_{\min}}^{a_j-\frac{T}{2}}\frac{\frac{\lambda_j t T}{2\pi}-k}{k\left(T+k-\frac{\lambda_j t T}{2\pi}\right)^2}\\
&\leqslant \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\left(\frac{\frac{\lambda_j t T}{2\pi}-k_{\min}}{k_{\min}\left(T+k_{\min}-\frac{\lambda_j t T}{2\pi}\right)^2} + \frac{4(T+2\Delta_j)}{(T-2\Delta_j)^2(2a_j-T)} + \int_{k=k_{\min}}^{a_j-\frac{T}{2}}\frac{\frac{\lambda_j t T}{2\pi}-k}{k\left(T+k-\frac{\lambda_j t T}{2\pi}\right)^2}dk\right)\\
&= \frac{\sin^2\left(\frac{\lambda_j t T}{2}\right)}{4}\left(\frac{\frac{\lambda_j t T}{2\pi}-k_{\min}}{k_{\min}\left(T+k_{\min}-\frac{\lambda_j t T}{2\pi}\right)^2} + \frac{4(T+2\Delta_j)}{(T-2\Delta_j)^2(2a_j-T)} + \frac{T}{T-\frac{\lambda_j t T}{2\pi}}\left(\frac{2}{T-2\Delta_j} - \frac{1}{T-\frac{\lambda_j t T}{2\pi}+k_{\min}}\right)\right.\\
&\quad\left. + \frac{\frac{\lambda_j t T}{2\pi}}{\left(T-\frac{\lambda_j t T}{2\pi}\right)^2}\ln\left(\frac{(2a_j-T)\left(T-\frac{\lambda_j t T}{2\pi}+k_{\min}\right)}{k_{\min}(T-2\Delta_j)}\right)\right) = O\left(\frac{\ln T}{T}\right).
\end{aligned} \tag{B11}
$$

This bound shows terms that diverge when $\lambda_j t/(2\pi)$ approaches 1, but the divergence is avoided by choosing $\max_j \lambda_j < 1$ and $t < 2\pi$. Indeed, $t$ is a free parameter of the algorithm, and $\max_j \lambda_j < 1$ can be achieved by a rescaling of the linear problem as discussed in Sec. II.

The other two sums in the right-hand side of (B10) are bounded following very similar computations. Finally, we obtain Eq. (20) for any value $0 < \lambda_j t/(2\pi) < 1$.

## APPENDIX C: COMPUTATION OF THE BOUND (21)

In this appendix we compute the upper and lower bounds in Eq. (21). These bounds show the dependence of the computational error on the algorithm parameters $T$ and $k_{\min}$. In particular, the error does not decrease with $T$ when $k_{\min} = O(T^0)$ which corresponds to a lower bound of the conditional number $\kappa' = tT/(2\pi k_{\min}) = O(T)$.

We start with upper bounding the error contribution (17) using the triangle inequality:

$$|\epsilon_{2,j}| = \lambda_j^2 \left| \sum_{k=k_{\min}}^{T-1} |\alpha_{k|j}^{\mathrm{var}}|^2 \left(\frac{tT}{2\pi k}\right)^2 - \frac{1}{\lambda_j^2} \right| \leqslant \sum_{k=0}^{k_{\min}-1} |\alpha_{k|j}^{\mathrm{var}}|^2 + \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2} \sum_{k=k_{\min}}^{T-1} \frac{\left|\frac{\lambda_j tT}{2\pi} - k\right| \left(\frac{\lambda_j tT}{2\pi} + k\right)}{k^2 \sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k\right)\right)}. \tag{C1}$$

The first sum is bounded by $O(\lambda_j tT)^{-1}$, as shown in Eq. (B4). In order to bound the second sum, we follow the same steps as in the previous appendix. For concreteness, we show only the case $\lambda_j t/(2\pi) < 1/2$, because the other case is very similar. Splitting the sum as in Eq. (B6) and recalling the notation (B5), the first contribution is the largest one and is bounded as follows:

$$\begin{aligned}
\frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2} \sum_{k=k_{\min}}^{a_j} \frac{\left|\frac{\lambda_j tT}{2\pi} - k\right| \left(\frac{\lambda_j tT}{2\pi} + k\right)}{k^2 \sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k\right)\right)} &\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4} \sum_{k=k_{\min}}^{a_j} \frac{k + \frac{\lambda_j tT}{2\pi}}{k^2 \left(\frac{\lambda_j tT}{2\pi} - k\right)} \\
&\leqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4} \left( \int_{k_{\min}}^{a_j} \frac{k + \frac{\lambda_j tT}{2\pi}}{k^2 \left(\frac{\lambda_j tT}{2\pi} - k\right)} dk + \frac{a_j + \frac{\lambda_j tT}{2\pi}}{\Delta_j a_j^2} + \frac{\frac{\lambda_j tT}{2\pi} + k_{\min}}{k_{\min}^2 \left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)} \right) \\
&= \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{4} \left( \frac{4\pi}{\lambda_j tT} \ln\left( \frac{a_j \left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)}{k_{\min}\Delta_j} \right) + \frac{a_j - k_{\min}}{k_{\min} a_j} + \frac{2a_j + \Delta_j}{\Delta_j a_j^2} + \frac{\frac{\lambda_j tT}{2\pi} + k_{\min}}{k_{\min}^2 \left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)} \right) \\
&\leqslant O\left(\frac{\ln(\lambda_j tT)}{\lambda_j tT}\right) + O\left(\frac{\kappa'}{tT}\right),
\end{aligned} \tag{C2}$$

where we have used $(a_j - k_{\min})/(k_{\min}(a_j)) = O(\kappa'/tT)$ in the last inequality. Using once again the same techniques, all the other terms coming from the splitting (B6) are bounded by orders $O(\ln(\lambda_j tT)/(\lambda_j tT))$. We then prove the upper bound in Eq. (21).

For the computation of the lower bound, we look again at the sum with $k \in [k_{\min}, a_j]$. Since all the terms are positive, we get a lower bound by neglecting all terms except $k = k_{\min}$:

$$\begin{aligned}
\frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{T^2} \sum_{k=k_{\min}}^{a_j} \frac{\left(\frac{\lambda_j tT}{2\pi} - k\right)\left(\frac{\lambda_j tT}{2\pi} + k\right)}{k^2 \sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k\right)\right)} &\geqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{k_{\min}^2 T^2} \frac{\left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)\left(\frac{\lambda_j tT}{2\pi} + k_{\min}\right)}{\sin^2\left(\frac{\pi}{T}\left(\frac{\lambda_j tT}{2\pi} - k_{\min}\right)\right)} \\
&\geqslant \frac{\sin^2\left(\frac{\lambda_j tT}{2}\right)}{\pi^2 k_{\min}^2} \frac{\frac{\lambda_j tT}{2\pi} + k_{\min}}{\frac{\lambda_j tT}{2\pi} - k_{\min}} \geqslant \sin^2\left(\frac{\lambda_j tT}{2}\right)\left(\frac{2\kappa'}{tT}\right)^2.
\end{aligned} \tag{C3}$$

All the other contributions of the splitting (B6) are orders $O[\ln(\lambda_j tT)/(\lambda_j tT)]$. We then get the lower bound in Eq. (21).


[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University, New York, 2010).

[2] G. Benenti, G. Casati, D. Rossini, and G. Strini, *Principles of Quantum Computation and Information: A Comprehensive Textbook* (World Scientific, Singapore, 2019).

[3] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (SFCS '94)* (IEEE Computer Society, Los Alamitos, CA, 1994), pp. 124–134.

[4] S. Aaronson and A. Arkhipov, in *STOC '11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2011), pp. 333–342.

[5] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, Theory Comput. **9**, 143 (2013).

[6] J. Watrous, in *Encyclopedia of Complexity and Systems Science*, edited by R. A. Meyers (Springer, New York, 2009), pp. 7174–7201.

[7] P. Arnault, P. Arrighi, S. Herbert, E. Kasnetsi, and T. Li, A typology of quantum algorithms, arXiv:2407.05178.

[8] S. Sagiroglu and D. Sinanc, in *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS)* (IEEE, 2013), p. 42.

[9] J. Fan, F. Han, and H. Liu, Challenges of big data analysis, Natl. Sci. Rev. **1**, 293 (2014).

[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).

[11] M. Morales, L. Pira, P. Schleich, K. Koor, P. Costa, D. An, A. Aspuru-Guzik, L. Lin, P. Rebentrost, and D. Berry, Quantum linear system solvers: A survey of algorithms and applications, arXiv:2411.02522.

[12] P. C. S. Costa, D. An, R. Babbush, and D. Berry, The discrete adiabatic quantum linear system solver has lower constant factors than the randomized adiabatic solver, Quantum **9**, 1887 (2025).

[13] P. C. S. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, Optimal scaling quantum linear-systems solver via discrete adiabatic theorem, PRX Quantum **3**, 040303 (2022).

[14] A. M. Dalzell, A shortcut to an optimal quantum linear system solver, arXiv:2406.12086.

[15] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019* (Association for Computing Machinery, New York, 2019), p. 193.

[16] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, Grand unification of quantum algorithms, PRX Quantum **2**, 040203 (2021).
[17] D. Camps, L. Lin, R. Van Beeumen, and C. Yang, Explicit quantum circuits for block encodings of certain sparse matrices, SIAM J. Matrix Anal. Appl. **45**, 801 (2024).
[18] A. Setty, Block encoding of sparse matrices via coherent permutation, arXiv:2508.21667.
[19] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[20] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).
[21] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026.
[22] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited, Proc. R. Soc. A **454**, 339 (1998).
[23] D. S. Abrams and S. Lloyd, Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, Phys. Rev. Lett. **83**, 5162 (1999).
[24] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation Contemp. Math. **305**, 53 (2002).
[25] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96* (Association for Computing Machinery, New York, 1993), p. 212.
[26] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. **79**, 325 (1997).
[27] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, Fortschr. Phys. **46**, 493 (1998).
[28] C. Cade, M. Folkertsma, S. Gharibian, R. Hayakawa, F. Le Gall, T. Morimae, and J. Weggemans, in *Proceedings of the 50th EATCS International Colloquium on Automata, Languages and Programming (ICALP 2023)* (Schloss Dagstuhl, Leibniz-Zentrum für Informatik, 2023), Vol. 9, pp. 32:1–32:19.
[29] Y. Cao, A. Daskin, S. Frankel, and S. Kais, Quantum circuit design for solving linear systems of equations, Mol. Phys. **110**, 1675 (2012).
[30] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakic, A. Aspuru-Guzik, and P. Walther, A two-qubit photonic quantum processor and its application to solving systems of linear equations, Sci. Rep. **4**, 6115 (2014).
[31] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig, Quantum linear systems algorithms: A primer, arXiv:1802.08227.
[32] B. Duan, J. Yuan, C.-H. Yu, J. Huang, and C.-Y. Hsieh, A survey on HHL algorithm: From theory to application in quantum machine learning, Phys. Lett. A **384**, 126595 (2020).
[33] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, Experimental realization of quantum algorithm for solving linear systems of equations, Phys. Rev. A **89**, 022313 (2014).
[34] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Natl. Bur. Stand. **45**, 255 (1950).
[35] N. J. Higham, FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation, ACM Trans. Math. Softw. **14**, 381 (1988).
[36] S.-C. T. Choi, C. C. Paige, and M. A. Saunders, MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems, SIAM J. Sci. Comput. **33**, 1810 (2011).
[37] H. Avron, A. Druinsky, and S. Toledo, Spectral condition-number estimation of large sparse matrices, Numer. Linear Algebra Appl. **26**, e2235 (2019).
[38] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. **270**, 359 (2007).
[39] B. D. W. and C. A. M., Quantum information and computation Quantum Inf. Comput. **12**, 29 (2012).
[40] D. W. Berry, R. Cleve, and S. Gharibian, Gate-efficient discrete simulations of continuous-time quantum query algorithms, arXiv:1211.4637.
[41] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *STOC '14: Proceedings of the 46th ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2014), p. 283.
[42] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, Phys. Rev. Lett. **114**, 090502 (2015).
[43] D. W. Berry, A. M. Childs, and R. Kothari, in *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS 2015)* (IEEE, New York, 2015), p. 792.
[44] D. Berry and L. Novo, Corrected quantum walk for optimal Hamiltonian simulation, Quant. Inf. Comp. **16**, 1295 (2016).
[45] D. Berry and L. Novo, Improved Hamiltonian simulation via a truncated Taylor series and corrections, Quant. Inf. Comp. **17**, 623 (2017).
[46] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, Phys. Rev. Lett. **118**, 010501 (2017).
[47] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).
[48] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).
[49] D. Porras and J. I. Cirac, Effective quantum spin systems with trapped ions, Phys. Rev. Lett. **92**, 207901 (2004).
[50] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).
[51] C. Monroe, W. C. Campbell, L.-M. Duan, Z.-X. Gong, A. V. Gorshkov, P. W. Hess, R. Islam, K. Kim, N. M. Linke, G. Pagano, P. Richerme, C. Senko, and N. Y. Yao, Programmable quantum simulations of spin systems with trapped ions, Rev. Mod. Phys. **93**, 025001 (2021).
[52] in *Quantum State Estimation*, edited by M. Paris and J. Řeháček, Lecture Notes in Physics Vol. 649 (Springer-Verlag, Berlin, 2004).
[53] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Quantum state tomography via compressed sensing, Phys. Rev. Lett. **105**, 150401 (2010).
[54] D. Gross, Recovering low-rank matrices from few coefficients in any basis, IEEE Trans. Inf. Theory **57**, 1548 (2011).
[55] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert, Quantum tomography via compressed sensing: Error bounds, sample complexity and efficient estimators, New J. Phys. **14**, 095022 (2012).
[56] H. J. J. Morrell, A. Zaman, and H. Y. Wong, A step-by-step HHL algorithm walkthrough to enhance understanding of

critical quantum computing concepts, IEEE Access **11**, 77117 (2023).

[57] P. Kaye and M. Mosca, *International Conference on Quantum Information* (Optica, Washington, DC, 2001), p. PB28.

[58] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv:quant-ph/0208112.

[59] L. K. Grover, Synthesis of quantum superpositions by quantum computation, Phys. Rev. Lett. **85**, 1334 (2000).

[60] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Transformation of quantum states using uniformly controlled rotations, Quantum Info. Comput. **5**, 467 (2005).

[61] A. N. Soklakov and R. Schack, Efficient state preparation for a register of quantum bits, Phys. Rev. A **73**, 012307 (2006).

[62] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).

[63] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, A divide-and-conquer algorithm for quantum state preparation, Sci. Rep. **11**, 6329 (2021).

[64] X.-M. Zhang, M.-H. Yung, and X. Yuan, Low-depth quantum state preparation, Phys. Rev. Res. **3**, 043200 (2021).

[65] S. Wang, Z. Wang, G. Cui, S. Shi, R. Shang, L. Fan, W. Li, Z. Wei, and Y. Gu, Fast black-box quantum state preparation based on linear combination of unitaries, Quantum Info. Proc. **20**, 270 (2021).

[66] J. Bausch, Fast black-box quantum state preparation, Quantum **6**, 773 (2022).

[67] B. D. Clader, A. M. Dalzell, N. Stamatopoulos, G. Salton, M. Berta, and W. J. Zeng, Quantum resources required to block-encode a matrix of classical data, IEEE Trans. Quantum Eng. **3**, 1 (2022).

[68] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T gates for dirty qubits in state preparation and unitary synthesis, Quantum **8**, 1375 (2024).

[69] A. Luis and J. Peřina, Optimum phase-shift estimation and the quantum description of the phase difference, Phys. Rev. A **54**, 4564 (1996).

[70] V. Bužek, R. Derka, and S. Massar, Optimal quantum clocks, Phys. Rev. Lett. **82**, 2207 (1999).

[71] D. W. Berry and H. M. Wiseman, Optimal states and almost optimal adaptive measurements for quantum interferometry, Phys. Rev. Lett. **85**, 5098 (2000).

[72] A. E. Rastegin, Relative error of state-dependent cloning, Phys. Rev. A **66**, 042304 (2002).

[73] A. E. Rastegin, Sine distance for quantum states, arXiv:quant-ph/0602112.

[74] T. W. Anderson, Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes, Ann. Math. Stat. **23**, 193 (1952).

[75] W. Hoeffding, Probability inequalities for sums of bounded random variables, J. Am. Stat. Assoc. **58**, 13 (1963).

[76] The spectral decomposition of $A$ is $A=\sum_l \lambda_l|u_l\rangle\langle u_l|$, and, using the Cauchy-Schwarz inequality, $|A_{ij}| = |\sum_l \langle e_i|u_l\rangle\lambda_l\langle u_l|e_j\rangle| \leqslant \sqrt{\sum_l |\langle e_i|u_l\rangle|^2\lambda_l}\sqrt{\sum_l |\langle e_j|u_l\rangle|^2\lambda_l} \leqslant \max_l\{\lambda_l\}$.

[77] *NIST Handbook of Mathematical Functions*, edited by D. M. Lozier, R. F. Boisvert, and C. W. Clark (Cambridge University, New York, 2010).

[78] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, Phys. Rev. Lett. **100**, 160501 (2008).

[79] X.-M. Zhang, T. Li, and X. Yuan, Quantum state preparation with optimal circuit depth: Implementations and applications, Phys. Rev. Lett. **129**, 230504 (2022).

[80] X.-M. Zhang and X. Yuan, Circuit complexity of quantum access models for encoding classical data, npj Quantum Inf. **10**, 42 (2024).

[81] X.-Q. Zhou, T. C. Ralph, P. Kalasuwan, M. Zhang, A. Peruzzo, B. P. Lanyon, and J. L. O'Brien, Adding control to arbitrary unknown quantum operations, Nat. Commun. **2**, 413 (2011).

[82] X.-Q. Zhou, P. Kalasuwan, T. C. Ralph, and J. L. O'Brien, Calculating unknown eigenvalues with a quantum algorithm, Nat. Photon. **7**, 223 (2013).

[83] N. Friis, V. Dunjko, W. Dür, and H. J. Briegel, Implementing quantum control for unknown subroutines, Phys. Rev. A **89**, 030303(R) (2014).

[84] https://github.com/MatiasGinzburg/Improvement-on-the-Convergence-in-Quantum-Linear-System-Solvers.

[85] F. Mezzadri, How to generate random matrices from the classical compact groups, arXiv:math-ph/0609050.

[86] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Phys. Rev. A **52**, 3457 (1995).