POLITECNICO DI TORINO
Repository ISTITUZIONALE

Extending the Step-by-Step HHL Algorithm Walkthrough: Advanced Concepts for Practical Quantum Computing Applications

Original
Extending the Step-by-Step HHL Algorithm Walkthrough: Advanced Concepts for Practical Quantum Computing Applications / Asinari, P., Piredda, M.M., Barletta, G., Fasano, M., Chiavazzo, E.. - In: IEEE ACCESS. - ISSN 2169-3536. - ELETTRONICO. - 14:(2026), pp. 105941-105959. [10.1109/ACCESS.2026.3712081]

Availability:
This version is available at: 11583/3013790 since: 2026-07-31T09:22:22Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/ACCESS.2026.3712081

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

09 November 2026

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 June 2026, accepted 1 July 2026, date of publication 10 July 2026, date of current version 17 July 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3712081

TUTORIAL

Extending the Step-by-Step HHL Algorithm Walkthrough: Advanced Concepts for Practical Quantum Computing Applications

PIETRO ASINARI[1,2], MATTEO MARIA PIREDDA[1], GIULIO BARLETTA[1], MATTEO FASANO[1], AND ELIODORO CHIAVAZZO[1,2]

[1]Dipartimento Energia, Politecnico di Torino, 10129 Turin, Italy
[2]Istituto Nazionale di Ricerca Metrologica, 10135 Turin, Italy

Corresponding author: Pietro Asinari (pietro.asinari@polito.it)

ABSTRACT Quantum computing has generated significant expectations within the scientific community. However, the development of noise-resilient quantum computers capable of addressing a broad class of practical problems is expected to require at least another decade. During this ongoing evolution, rigorous step-by-step tutorials are essential not only to clarify fundamental concepts but also to disseminate them beyond the core quantum computing community to the wider engineering audience interested in applications. In many engineering fields, the efficient solution of linear systems of equations is a central computational task. For this reason, although it remains an open question whether the Harrow–Hassidim–Lloyd (HHL) algorithm will ultimately play a decisive role in future fault-tolerant application-scale quantum (FASQ) computers, clear and rigorous tutorials on this paradigmatic algorithm remain of fundamental importance. While well-structured introductions to elementary implementations of HHL already exist, certain simplifying assumptions—appropriate for pedagogical clarity—may limit their extension to more general or application-oriented settings. In this work, the HHL algorithm is presented through a detailed analytical treatment, followed by a complete example including a generic module for eigenvalue inversion. A corresponding Jupyter Notebook implementation in Qiskit is provided to support practical understanding. After working through the material, readers are expected to gain a clearer appreciation of how quantum computing concepts operate in practice and why quantum approaches are of interest for combinatorial optimization and machine learning problems.

INDEX TERMS Harrow-Hassidim-Lloyd (HHL) quantum algorithm, quantum Fourier transform (QFT), inverse quantum Fourier transform (IQFT), quantum phase estimation (QPE), linear system problem (LSP), quantum education.

I. INTRODUCTION

Quantum computing has generated significant expectations within the scientific community. To date, however, only theoretical evidence suggests that quantum computers may provide substantial advantages over classical hardware in specific domains, including: (i) the simulation of quantum physics and chemistry, (ii) the breaking of currently deployed public-key cryptographic schemes, and (iii) optimization problems arising in finance and logistics [1]. Beyond chemistry, cryptography, and finance, another area attracting considerable attention is (iv) machine learning [2]. These applications share a common feature: they involve combinatorial problems that are challenging for classical computers but may benefit from quantum computation, where multiple solution components can be processed in superposition. Nevertheless, computational challenges in engineering extend well beyond combinatorial tasks and could, in principle, also benefit from quantum approaches once fault tolerance—or at least effective error

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

© 2026 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

mitigation—is achieved. In particular, the numerical solution of partial differential equations plays a central role in many engineering applications. Such problems typically require the solution of high-dimensional, sparse linear systems, commonly referred to as the linear system problem (LSP).

Quantum algorithms for solving linear systems of equations, commonly referred to as Quantum Linear System Algorithms (QLSAs), constitute a fundamental building block for quantum-enhanced engineering simulations. For near-term quantum hardware, variational quantum algorithms (VQAs) [3] are typically the preferred approach. VQAs are hybrid methods that combine quantum and classical computational resources: a parameterized quantum circuit is executed on quantum hardware to prepare and measure quantum states, while a classical optimizer iteratively updates the circuit parameters based on measurement outcomes. VQAs are particularly well suited to current NISQ devices because they employ relatively shallow circuits, exhibit a degree of resilience to noise, and delegate the computationally intensive optimization process to classical processors [4], [5], [6], [7]. Despite the current hardware limitations, the canonical QLSA remains the Harrow–Hassidim–Lloyd (HHL) algorithm [8] along with its subsequent refinements [9]. Under assumptions on sparsity, condition number, efficient Hamiltonian simulation, state preparation, and output access, HHL can provide exponential scaling advantages for specific linear-system tasks. However, it requires deep quantum circuits and thus assumes a fault-tolerant quantum computer. Here, fault-tolerant refers to a fully error-corrected quantum device capable of executing long and precise quantum circuits without the computation being compromised by noise.

The development of noise-resilient quantum computers capable of addressing a broad class of practical problems is expected to require at least another decade [10]. However, the existence of noisy intermediate-scale quantum (NISQ) computers stimulated algorithms to leverage the available quantum resources. A thorough summary of NISQ computational paradigms and algorithms is provided in this review [11]. Among these developments, hybrid HHL algorithms have been proposed that reduce the circuit depth of the original HHL algorithm by exploiting classical feed-forward of information obtained from the quantum phase estimation stage [12], [13]. During the transitional phase from current NISQ to fully error-corrected computers, rigorous, step-by-step tutorials are essential not only to clarify fundamental concepts but also to extend their accessibility beyond the core quantum computing community to the broader engineering audience interested in applications. For this reason, although it remains an open question whether the HHL algorithm will ultimately play a decisive role in future fault-tolerant application-scale quantum (FASQ) computers, clear and rigorous tutorials on this paradigmatic algorithm remain of primary importance. Numerous HHL tutorials are already available: some focus on the underlying physical and mathematical foundations (e.g., [14], [15]), while others are presented within quantum computing software frameworks (e.g., [16], [17], [18]). However, even well-structured introductions to elementary implementations often rely on simplifying assumptions—appropriate for pedagogical clarity—that may limit their extension to more general or application-oriented scenarios. For instance, the eigenvalue inversion module required by HHL is sometimes constructed assuming prior knowledge of the eigenvalues [14], with more systematic approaches (e.g., [19]) deferred to advanced treatments.

In this work, the HHL algorithm is presented through a detailed analytical treatment, followed by a complete example that can be extended to more general cases without restrictive assumptions. The example addresses an engineering problem relevant to thermal science, previously investigated using a variational quantum approach for comparison [7]. A corresponding Jupyter Notebook implementation in Qiskit (Quantum Information Software Kit) [16] is made available [20] to support a practical understanding of the implementation details. The notebook has been tested with Python 3.12 and Qiskit 2.4.1. Minor adjustments may be required in the future as the Qiskit API evolves. After working through the example, readers are expected to develop a clearer appreciation of how quantum computing concepts operate in practice and why quantum approaches are attracting increasing interest, particularly in combinatorial optimization and machine learning. In particular, an introduction to quantum machine learning with a bottom-up view can be found in [21].

A. HOW TO USE THIS PAPER

This work is intended as a follow-up to the well-structured tutorial by Zaman et al. [14] on the same topic. Our goal is for the present contribution to complement that tutorial, providing a natural extension that bridges foundational explanations with more advanced and application-oriented aspects. The two tutorials are designed to be complementary; accordingly, we adopt a consistent methodological framework to ensure coherence and continuity between them. Although this document is self-contained and can be read independently by readers already familiar with the basic concepts, we nevertheless recommend consulting the tutorial by Zaman et al. [14] first, as it provides essential background that facilitates a deeper understanding of the material presented here. For interested readers with an industrial engineering background, some basic quantum computing concepts are also discussed in the appendices of [7].

Section II introduces the linear system problem investigated in this work, arising in thermal science from the solution of the heat conduction equation in a periodic one-dimensional domain. This problem represents a paradigmatic test case for many engineering applications and has previously been studied using a variational quantum approach for comparison [7].

Section III presents a step-by-step analytical discussion of the HHL algorithm, adopting the same notation and

labeling of quantum states used in [14] to ensure consistency, and avoiding simplifying approximations. A corresponding Jupyter Notebook implementation in Qiskit (Quantum Information Software Kit) [16] is made available [20] to facilitate a practical understanding of the implementation details. Selected technical aspects that would otherwise interrupt the flow of the main discussion are treated in the Appendices, including real data loading/encoding (Appendix A), the phase kickback mechanism (Appendix B) and the error analysis (Appendix C), all of which are explicitly referenced in the main text.

For readers already familiar with HHL through commonly adopted simplified derivations, Section IV revisits the algorithm under standard simplifying assumptions, thereby providing a direct comparison with the more general treatment developed in this work.

II. LINEAR SYSTEM PROBLEM (LSP)

Let us consider the one-dimensional heat conduction equation, as a simple yet paradigmatic test case for engineering applications [7], namely

$$\frac{\partial T}{\partial t} = D\,\frac{\partial^2 T}{\partial z^2}, \tag{1}$$

with the function $T = T(z,t)$ being the local temperature, z the space coordinate, t the time, and the positive coefficient D the thermal diffusivity of the medium. Let us consider a constant diffusivity, a given initial profile $T(z,0)$ and the periodic spatial boundary condition. This problem can be solved analytically using the Fourier transform and it is usually trivial for most of the current classical numerical techniques.

Let us solve the previous equation by the classical finite-difference (FD) method, which consists in solving differential equations by approximating derivatives with finite differences. Both the spatial domain and time domain are discretized by a regular mesh: the unknown function T is evaluated at the generic l-th mesh node and at the τ-th time step, where both l and τ belong to $\mathbb{N}^+$, namely $T_l^\tau = T(z_l, t_\tau)$ where $z_l = l\,\Delta z$ with $0 \le l \le N-1$ and $t_\tau = \tau\,\Delta t$ with $0 \le \tau \le N_t$ ($\tau = 0$ identifies the given initial profile). The quantities Δz and Δt are the spatial and temporal partitions of the grid, while N is the number of space mesh nodes and N_t is the number of time steps. Let us use a fully implicit FD scheme that yields the stability of the solution for arbitrary diffusivity of the equation and the grid size:

$$\frac{T_l^{\tau+1} - T_l^\tau}{\Delta t} = D\,\frac{T_{l-1}^{\tau+1} - 2\,T_l^{\tau+1} + T_{l+1}^{\tau+1}}{\Delta z^2}. \tag{2}$$

The previous formula can be reformulated as

$$-r\,T_{l-1}^{\tau+1} + (1+2\,r)\;T_l^{\tau+1} - r\,T_{l+1}^{\tau+1} = T_l^\tau, \tag{3}$$

where $r = D\,\Delta t/\Delta z^2$ is the (dimensionless) numerical Fourier number. Let us define a new operator $\hat{C}$ as

$$\hat{C} := \begin{bmatrix} (1+2\,r) & -r & \dots 0 \dots & -r \\ -r & (1+2\,r) & -r & \dots 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 \dots & -r & (1+2\,r) & -r \\ -r & \dots 0 \dots & -r & (1+2\,r) \end{bmatrix}, \tag{4}$$

which can be used to formulate a linear system of equations which is consistent with Eq. (2). Because each row of $\hat{C}$ contains the diagonal real entry $1+2\,r$ and off-diagonal real entries $-r$, the Gershgorin circle theorem (1931) states that all eigenvalues of $\hat{C}$ are real and lie within discs centered at $1+2\,r$ with radius $2\,r$. Therefore the generic m-th eigenvalue λ_m^C is contained in the interval

$$\lambda_m^C \in [1,\, 1+4\,r]\,. \tag{5}$$

The operator $\hat{C}$ defined by Eq. (4) can be used to express Eq. (2) in compact form

$$\hat{C}\,\vec{T}^{+} = \vec{T}, \tag{6}$$

where $\vec{T}^{+}$ stands for $\vec{T}^{\tau+1} = \left(T_0^{\tau+1}, T_1^{\tau+1}, \dots, T_{N-1}^{\tau+1}\right)^T$ and $\vec{T}$ stands for $\vec{T}^{\tau} = \left(T_0^{\tau}, T_1^{\tau}, \dots, T_{N-1}^{\tau}\right)^T$. Clearly the inverse of operator $\hat{C}$ can be used as a time-progress operator for the temperature profile subject to heat conduction, namely

$$\vec{T}^{+} = \vec{T}(t+\Delta t) = \hat{C}^{-1}\,\vec{T}, \tag{7}$$

which can be also generalized by $\tau \in \mathbb{N}^+$ in the following formula

$$\vec{T}(t+\tau\,\Delta t) = (\hat{C}^{-1})^\tau\;\vec{T}^{t=0}. \tag{8}$$

In the following sections, for the sake of simplicity and without loss of generality, we will focus on $\tau = 1$ and hence on Eq. (7) only.

In order to define a proper LSP, Eq. (6) must be coupled with an initial temperature profile. The initial temperature profile $\vec{T}^{t=0}$ can excite different eigenmodes of the heat conduction equation. Consequently, the choice of the initial temperature profile—and thus its complexity—is an integral part of defining the test case. For the sake of simplicity, we consider an initial temperature profile $\vec{T}^{t=0}$ defined by a single harmonic, namely

$$T_l^{t=0} = T\,(l\,\Delta z, 0) = 1 + \frac{1}{2}\,\sin\left[\frac{2\pi}{N}\,(l+1)\right]. \tag{9}$$

The previous profile can be considered an input of the test case, while the output is given by applying the time-progress operator $\hat{C}^{-1}$ given by Eq. (7) to the input profile $\vec{T}^{t=0}$. In other words, a single time step is performed to update the initial temperature profile $\vec{T}^{t=0}$ of the target problem. The expected results can be anticipated by referring to the results reported in Fig. 5, which can be also helpful for visualizing the problem setup. In the following sections, this

simple test case is solved by the HHL algorithm. It is worth noticing that the reported example is intentionally small and analytically transparent; its purpose is pedagogical rather than to demonstrate quantum advantage over classical finite-difference solvers.

III. HARROW-HASSIDIM-LLOYD (HHL) ALGORITHM

A. OVERVIEW OF THE HHL ALGORITHM

Let us consider in this section the HHL algorithm [8] which, to date, can be considered one of the most promising quantum algorithms for solving linear systems on future, fault-tolerant quantum computers. HHL is theoretically appealing because it offers an exponential quantum speedup under well-defined assumptions—specifically, when the matrix is sparse, well-conditioned, and efficiently representable [8], with potential impact also on industrial applications [22]. In spite of the promising features, the practical implementation of HHL on current NISQ devices remains severely limited. The algorithm requires deep circuits involving controlled rotations, quantum phase estimation, and accurate eigenvalue inversion—operations highly sensitive to gate noise, decoherence, and restricted circuit depth. Thus, while HHL stands as a theoretically powerful algorithm with strong asymptotic promises, its practical use is postponed to the era of large-scale, error-corrected quantum computers. Meanwhile, it is worth the effort to investigate the applicability of the HHL algorithm in solving practical problems by using classical High Performance Computing (HPC) facilities [22]. The main goal of this document is to provide a rigorous analysis of the HHL algorithm and to discuss its details in a clear, step-by-step manner. In this work, only ideal (statevector) simulations are reported, as the practical implementation of HHL on current NISQ devices remains severely constrained, particularly for problems beyond very small-scale instances.

First of all, the main difference between HHL and other methods is that the former requires extra qubits for approximating the solution. This is the case because HHL must temporarily store and process the eigenvalues of the matrix of the target problem and perform controlled operations that depend on those eigenvalues. These extra qubits hold intermediate quantum information, which are essential for implementing the inverse of the matrix of the linear system coherently on a quantum state. Therefore, in addition to the n qubits that encode the numerical solution at the mesh nodes (with $N = 2^n$), HHL requires additional n_c "clock" qubits and one extra ancilla qubit. Therefore, the total number of qubits required by the algorithm is $n+n_c+1$. Clearly, there is a noticeable overhead for small systems when $n \sim (n_c + 1)$, but this becomes negligible when $n \gg (n_c+1)$. We can regard the n qubits used for the computational mesh as the input/output register, since they are employed to encode both the constant input terms of the linear system and the corresponding solution values. A quantum register is a collection of qubits that together form the basic unit of quantum memory or state representation in a quantum computer. In the following, for the sake of simplicity, the input/output register will be indicated as input register only, even though it will be used to store the solution values as well. Similarly, the n_c qubits form the clock register and the ancilla qubit constitutes the ancilla register. Hence, the HHL algorithm uses three quantum registers, which will be conventionally listed in the above order. Within each register, the corresponding computational basis is defined by a binary representation, which uses the usual mathematical notation to represent the bit strings, i.e., $\beta_{n-1} \dots \beta_1 \beta_0$, where each digit $\beta_i \in 0, 1$, as in conventional binary encoding. In order to clarify the meaning of the qubits, it is also possible to add a subscript or superscript which specifies the register they belong to (omitting the one for input register), namely

$$|\cdots\rangle\,|\cdots\rangle_c\,|\cdots\rangle_a = |\beta_{n-1} \dots \beta_1 \beta_0\, \beta^c_{n_c-1} \dots \beta^c_1 \beta^c_0\, \beta^a_0\rangle\,, \tag{10}$$

where the three registers are separable at the input and output, but not necessarily in the intermediate steps of the algorithm when they exchange/share information.

Returning to the main algorithm, the complete HHL circuit corresponding to the problem introduced in Section II is shown in Fig. 1. The HHL algorithm consists of two principal components: (i) the Quantum Phase Estimation (QPE) module and (ii) the eigenvalue inversion module. These components are arranged into four logical stages:

- *Initial state preparation*, describing the evolution from state $|\Psi_0\rangle$ to $|\Psi_1\rangle$ (following the notation of [14]), detailed in Section III-B;
- *QPE module*, describing the evolution from $|\Psi_1\rangle$ to $|\Psi_4\rangle$, presented in Section III-C;
- *Eigenvalue inversion module*, describing the evolution from $|\Psi_4\rangle$ to $|\Psi_6\rangle$, discussed in Section III-D;
- *Inverse QPE module*, describing the evolution from $|\Psi_6\rangle$ to the final state $|\Psi_9\rangle$, presented in Section III-E.

B. INITIAL STATE PREPARATION

In the HHL algorithm, the matrix associated with the linear system problem (LSP) must be Hermitian [8], [14]. When the LSP is formulated in terms of real-valued quantities, this requirement reduces to the condition that the system matrix be symmetric. Consequently, in the real case, HHL can directly address only LSPs characterized by symmetric matrices. If this condition is not satisfied, a non-Hermitian matrix can be embedded into a larger Hermitian matrix by doubling the dimension of the problem [14], which effectively requires one additional qubit in the quantum implementation. For a more advanced treatment of linear systems with non-Hermitian matrices, readers are referred to [23]. The Hermiticity requirement is naturally satisfied in the problem considered here, owing to its one-dimensional formulation and the adoption of periodic boundary conditions, which result in a symmetric discretized operator.

The first step of quantum computing is to identify the quantum state where to store the relevant information by

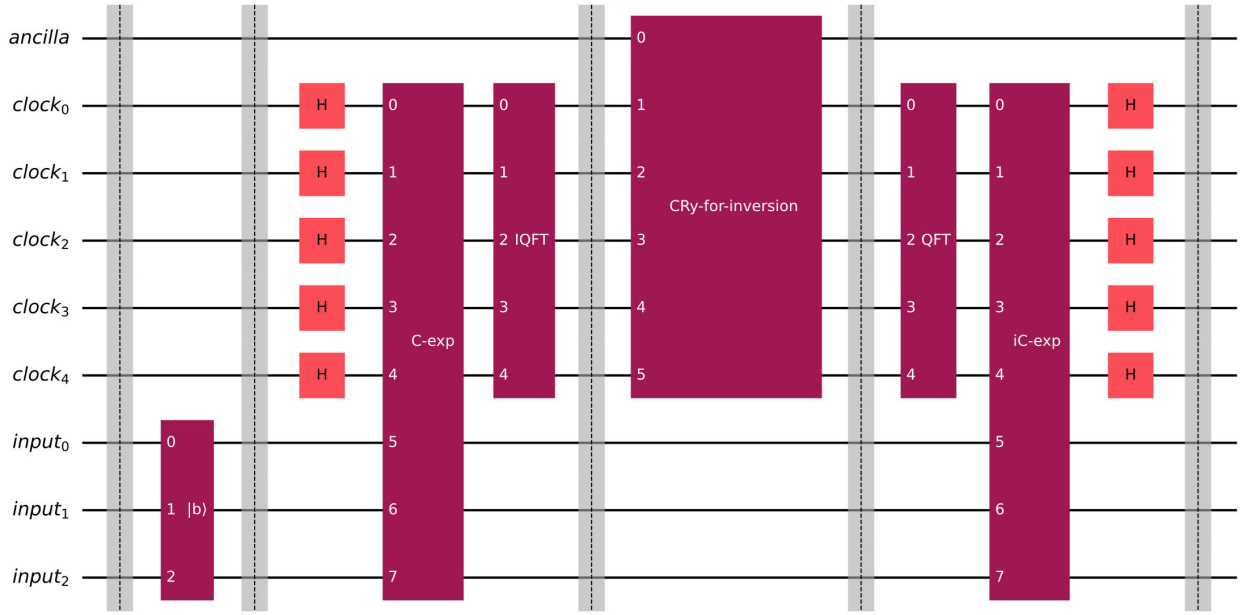

FIGURE 1. Circuit ideally implementing HHL algorithm ($n = 3$ input qubits, $n_c = 5$ clock qubits and 1 ancilla qubit) for solving the problem discussed in the previous sections. "C-exp" stands for controlled exponentiation (or controlled Hamiltonian evolution), while "iC-exp" stands for its logical inverse. "CRy-for-inversion" stands for controlled rotation for eigenvalue inversion. Since this circuit does not include any measurements, it is suitable only for ideal (statevector) simulations.

normalizing Eq. (6), because a discrete quantum state $|\psi\rangle$ is normalized, namely $\langle\psi|\psi\rangle = 1$. Let us define the quantum state $|b\rangle$ for mapping the initial temperature profile, namely

$$|b\rangle = \frac{1}{\sqrt{\vec{T}\cdot\vec{T}}}\vec{T}, \tag{11}$$

where, by construction, $\langle b|b\rangle = 1$. Similarly, let us proceed with the quantum state $|x\rangle$ for mapping the updated temperature profile at the new time step, namely

$$|x\rangle = \frac{1}{\sqrt{\vec{T}^{+}\cdot\vec{T}^{+}}}\vec{T}^{+}, \tag{12}$$

where the same normalization holds. Introducing these definitions into Eq. (6) yields

$$\hat{A}\,|x\rangle = |b\rangle\,, \tag{13}$$

where $\hat{A} = f\,\hat{C}$ and the normalization factor f is given by

$$f := \sqrt{\frac{\vec{T}^{+}\cdot\vec{T}^{+}}{\vec{T}\cdot\vec{T}}}. \tag{14}$$

The problem is that the normalization factor is not known at the beginning of the numerical procedure because it depends on the solution $\vec{T}^{+}$, which derives from solving the linear system of equations. To avoid circular dependencies, the matrix $\hat{A}$ may be introduced to rationalize the formulation of the problem; however, the HHL algorithm must be constructed solely in terms of the original matrix $\hat{C}$. Once the solution state $|x\rangle$ has been obtained, it can be appropriately rescaled to recover the updated temperature profile $\vec{T}^{+}$. This rescaling step exploits the energy conservation property of the system, which ensures that the average temperature of the updated profile $\vec{T}^{+}$ coincides with that of the initial profile $\vec{T}$, a quantity that is known *a priori*.

The first step consists in preparing the input register to have the amplitudes corresponding to the coefficients of $|b\rangle$. This can be done by many techniques designed for data loading/encoding in quantum circuits. Most quantum computing software frameworks (e.g., [16], [18]) provide dedicated routines for state preparation. Some algorithms rely on divide-and-conquer strategies for efficiently preparing amplitude-encoded states, exploiting hierarchical structures such as segment trees to reduce state-preparation costs [24]. Some details about the latter approach are reported in Appendix A.

These approaches require a number of classical input parameters that scales exponentially with the number of qubits, namely $\sim N = 2^n$ (up to normalization). Consequently, for very large problem sizes, the classical preprocessing and data transfer requirements become prohibitively demanding. For large-scale applications, it is therefore essential to minimize the amount of classical data passed to the quantum processor and instead generate the required quantum state internally, leveraging problem structure and compact parameterizations implemented directly on the quantum device. Hence, data encoding remains an active and challenging research area in quantum computing [25].

Following the same notation of [14], starting with $|\Psi_0\rangle := |0\rangle^{\otimes n}\,|0\rangle_c^{\otimes n_c}\,|0\rangle_a$ and applying a proper procedure for data

encoding yields

$$|\Psi_1\rangle := |b\rangle\ |0\rangle_c^{\otimes n_c}\ |0\rangle_a\,. \tag{15}$$

C. QUANTUM PHASE ESTIMATION (QPE)

The non-trivial starting point of the HHL algorithm is the Quantum Phase Estimation (QPE) algorithm [26]. The QPE algorithm is a quantum algorithm used to estimate the phase associated with an eigenvalue of a given unitary operator.

In order to understand what is the quantum phase, let us recall the spectrum of the eigenvalues of the numerical finite-difference procedure for the heat conduction equation given by Eq. (5) in Section II, namely $1 \leq \lambda_m^C \leq (1+4\,r)$, where λ_m^C is the m-th eigenvalue of the operator $\hat{C}$ given by Eq. (4). The parameter $r = D\,\Delta t/\Delta z^2$ is the (dimensionless) numerical Fourier number. It should increase during mesh refinement in order to keep Δt constant; namely, $r \sim 1/\Delta z^2$. Otherwise, time stepping would require an impractically large number of iterations on a quantum computer with a significant number of qubits. Therefore it makes sense to normalize the evolution matrix as

$$\hat{\Gamma} := \frac{1}{1+4\,r}\,\hat{C}. \tag{16}$$

The normalized evolution matrix $\hat{\Gamma}$ is real and symmetric. Hence it can be expressed by spectral decomposition, using its eigenvectors $|\phi_m^\Gamma\rangle$ and its eigenvalues λ_m^Γ, namely

$$\hat{\Gamma} = \sum_{m=0}^{N-1} \lambda_m^\Gamma\ |\phi_m^\Gamma\rangle\,\langle\phi_m^\Gamma| = \sum_{j\in\{0,\,1\}^n} \lambda_j^\Gamma\ |\phi_j^\Gamma\rangle\,\langle\phi_j^\Gamma|\,, \tag{17}$$

where j is the bit string corresponding to m.

Before proceeding, let us briefly recall some basics of binary encoding, which are relevant to both classical and quantum computing [26]. To represent a number in binary, every digit has to be either 0 or 1 (as opposed to decimal numbers where a digit can take any value from 0 to 9). In both cases, the mathematical notation used to write binary numbers or quantum states usually implies that the least significant bit (LSB) is written at the rightmost position. A decimal integer $m \equiv m_{(10)}$ can be decomposed in terms of a binary number, which is represented by an equivalent bit string $j_{(2)} = \beta_{n-1}\beta_{n-2}\ldots\beta_1\beta_0$, where $m \equiv m_{(10)} = \sum_{b=0}^{n-1} \beta_b\,2^b$ and $\beta_b \in \{0,1\}$ is the value of the b-th computational unit in a binary computational system with n bits. Because of the previous normalization, now the following relation holds

$$\frac{1}{1+4\,r} \leq \lambda_j^\Gamma \leq 1. \tag{18}$$

The definitions provided in Eqs. (11), (12) and (16) allow the system of linear equations in Eq. (6) to be recast as $g\,\hat{\Gamma}\ |x\rangle = |b\rangle$, where $g = f\,(1+4\,r)$ and f is given by Eq. (14). In particular, the input $|b\rangle$ can be also expressed in the basis formed by the eigenvectors of $\hat{\Gamma}$, such that

$$|b\rangle = \sum_{j\in\{0,\,1\}^n} b_j\,|j\rangle = \sum_{j\in\{0,\,1\}^n} b_j^\Gamma\ |\phi_j^\Gamma\rangle\,, \tag{19}$$

as well as the solution

$$|x\rangle = \sum_{j\in\{0,\,1\}^n} x_j^\Gamma\ |\phi_j^\Gamma\rangle\,. \tag{20}$$

Recalling that the eigenvectors of the matrix $\hat{\Gamma}$ form an orthonormal basis, namely $\langle\phi_j^\Gamma\,|\phi_k^\Gamma\rangle = \delta_{jk}$, yields

$$x_j^\Gamma = \frac{b_j^\Gamma}{g\,\lambda_j^\Gamma}. \tag{21}$$

Since $\langle x|x\rangle = 1$ by construction, then $\sum_j |x_j^\Gamma|^2 = 1$ holds. The last relations allow us to compute g, namely

$$g = \sqrt{\sum_j \left|\frac{b_j^\Gamma}{\lambda_j^\Gamma}\right|^2}, \tag{22}$$

and consequently

$$x_j^\Gamma = \frac{1}{\sqrt{\sum_j \left|b_j^\Gamma/\lambda_j^\Gamma\right|^2}}\,\frac{b_j^\Gamma}{\lambda_j^\Gamma}. \tag{23}$$

In essence, Eq. (21) reveals that the HHL algorithm is a step-by-step procedure for computing the coefficients x_j^Γ (and hence $|x\rangle$), by computing the expansion coefficients b_j^Γ and inverting the eigenvalues λ_j^Γ.

The last comment is the proper beginning of the QPE algorithm. First of all, Hadamard gates are applied to the clock qubits to create a superposition of the clock qubits, namely

$$|\Psi_2\rangle := |b\rangle\ \frac{1}{\sqrt{N_c}}\Big(\,|0\rangle + |1\rangle\,\Big)^{\otimes n_c}\ |0\rangle_a\,, \tag{24}$$

where $N_c = 2^{n_c}$. This superposition can be further expanded explicitly as

$$\begin{aligned}|\Psi_2\rangle &= |b\rangle\ \frac{1}{\sqrt{N_c}}\Big(\,|0\rangle + |1\rangle\,\Big)^{\otimes n_c}\ |0\rangle_a \\ &= |b\rangle\ \frac{1}{\sqrt{N_c}}\sum_{k\in\{0,1\}^{n_c}} |k\rangle_c\ |0\rangle_a\,,\end{aligned} \tag{25}$$

where k is again a binary number and $|k\rangle$ is a generic state of the computational basis of the clock register. Please note that the number N_c of the states of the clock register typically differs from the number N of the states of the input register. The previous system quantum state is still separable in its constituent registers, namely

$$|\Psi_2\rangle = \left(\sum_{j\in\{0,\,1\}^n} b_j^\Gamma\ |\phi_j^\Gamma\rangle\right) \otimes \left(\frac{1}{\sqrt{N_c}}\sum_{k\in\{0,\,1\}^{n_c}} |k\rangle_c\right) \otimes |0\rangle_a\,, \tag{26}$$

where we used Eq. (19). In the following, we will sometimes decompose the binary number k in a string of bits, namely

$$k := k_{(2)} = \beta_{n_c-1}^c \ldots \beta_1^c\beta_0^c, \tag{27}$$

where the generic clock bit is β_q^c. The corresponding decimal number is

$$k_{(10)} = \sum_{q=0}^{n_c-1} \beta_q^c \, 2^q. \tag{28}$$

The next step is the controlled exponentiation (or controlled Hamiltonian evolution) shown in Fig. 2. The QPE extracts eigenvalues (as quantum phases defined below), but only of a unitary operator. We can define the unitary operator $\hat{U}$ as the evolution determined by the Hamiltonian $\hat{\Gamma}$ we are interested in, namely

$$\hat{U} = e^{i\,\hat{\Gamma}\,\varphi} = \sum_{s=0}^{\infty} \frac{1}{s!} (i\,\hat{\Gamma}\,\varphi)^s = \sum_{j\in\{0,\,1\}^n} e^{i\,\lambda_j^{\Gamma}\,\varphi} \, |\phi_j^{\Gamma}\rangle \langle \phi_j^{\Gamma}| \,, \tag{29}$$

where φ is a fictitious evolution time (which also indirectly explains why the second register is called "clock"). Before proceeding, it is important to emphasize a key aspect of the HHL algorithm. HHL assumes that, given a sparse Hermitian matrix $\hat{\Gamma}$, the corresponding unitary time-evolution operator $e^{i\hat{\Gamma}\varphi}$ can be implemented efficiently on a quantum computer. This assumption is nontrivial and relies on the availability of efficient Hamiltonian-simulation techniques, such as Trotter–Suzuki product formulas, linear-combination-of-unitaries (LCU) methods, and qubitization-based approaches [26], [27]. The computational cost of implementing $e^{i\hat{\Gamma}\varphi}$ contributes to the overall complexity of the HHL algorithm. In this tutorial, we focus on the subsequent stages of HHL and assume that the required Hamiltonian evolution can be realized efficiently.

For reasons that will be clarified later, we assume

$$\varphi = 2\pi \left(\frac{N_c - 1}{N_c} \right). \tag{30}$$

A unitary operator has eigenvalues of modulus 1, which can be written as $e^{i2\pi\phi_j}$, where ϕ_j is the j-th quantum phase. Quantum phases are elusive concepts because multiplying an entire quantum state by $e^{i2\pi\phi_j}$ does not change measurement outcomes. However, the HHL algorithm, and QPE in particular, uses quantum phases of the input register to compute the eigenvalues we are interested in. It is worth noting that $\hat{U}$ is also diagonal in the basis defined by the eigenvalues of $\hat{\Gamma}$, namely

$$\hat{U} = \sum_{j\in\{0,\,1\}^n} e^{i2\pi\phi_j} \, |\phi_j^{\Gamma}\rangle \langle \phi_j^{\Gamma}| \,. \tag{31}$$

Comparing the previous expression with Eq. (29) allows to derive the relation between quantum phases and original eigenvalues as $2\pi\phi_j = \lambda_j^{\Gamma}\varphi$. Using the assumption given by Eq. (30) implies

$$\phi_j = \lambda_j^{\Gamma} \left(\frac{N_c - 1}{N_c} \right). \tag{32}$$

It is also trivial to compute the eigenvalues of the previous operator because

$$\hat{U} \, |\phi_j^{\Gamma}\rangle = e^{i2\pi\phi_j} \, |\phi_j^{\Gamma}\rangle = e^{i\lambda_j^{\Gamma}\varphi} \, |\phi_j^{\Gamma}\rangle \,. \tag{33}$$

For the powers of the unitary operator $\hat{U}$, similar relations hold. The powers of the unitary operator prove to be very useful. The idea is to use different powers of $\hat{U}$ in order to highlight the specific behavior of different quantum phases ϕ_j (and hence of different eigenvalues). We want to operate on the input register differently, depending on the clock register. Let us reformulate Eq. (26) as

$$|\Psi_2\rangle = \frac{1}{\sqrt{N_c}} \sum_{j\in\{0,\,1\}^n} b_j^{\Gamma} \sum_{k\in\{0,\,1\}^{n_c}} |\phi_j^{\Gamma}\rangle \, |k\rangle_c \, |0\rangle_a \,, \tag{34}$$

because $|\phi_j^{\Gamma}\rangle$ does not depend on k. Let us proceed in the following way: (i) first, we factorize its binary representation as $k = \beta_{n_c-1}^c \ldots \beta_1^c \beta_0^c$; (ii) second, we use β_q^c to perform some operations on $|\phi_j^{\Gamma}\rangle$. In particular, if $\beta_q^c = 1$ then we apply $\hat{U}^{2^q}$ to $|\phi_j^{\Gamma}\rangle$, otherwise, if $\beta_q^c = 0$, then nothing happens. It is possible to simplify the last statement by saying that, for every bit of the string $k = \beta_{n_c-1}^c \ldots \beta_1^c \beta_0^c$, one applies $\hat{U}^{\beta_q^c 2^q}$ on $|\phi_j^{\Gamma}\rangle$, because $\hat{U}^0 = I$. Since β_q^c depends on k, we can imagine to define the following operator (of Hamiltonian evolution)

$$E_V = \prod_{q=0}^{n_c-1} \hat{U}^{\beta_q^c(k)\,2^q} = \hat{U}^{\sum_{q=0}^{n_c-1} \beta_q^c(k)\,2^q} = \hat{U}^{k_{(10)}}. \tag{35}$$

Applying the previous operator to $|\phi_j^{\Gamma}\rangle$ yields

$$E_V \, |\phi_j^{\Gamma}\rangle = \hat{U}^k \, |\phi_j^{\Gamma}\rangle = e^{i2\pi\phi_j k_{(10)}} \, |\phi_j^{\Gamma}\rangle \,, \tag{36}$$

where we used Eq. (33) and Eq. (28). The operator E_V depends on the clock register $|k\rangle_c$ as it is clear in Eq. (35). Let us define the following controlled operator for the whole quantum system

$$\mathrm{CE}_V = \left(\sum_{k\in\{0,\,1\}^{n_c}} \hat{U}^k \otimes |k\rangle_c \langle k|_c \right) \otimes I_a, \tag{37}$$

and let us apply it to the system quantum state given by Eq. (34) which yields

$$\begin{aligned} |\Psi_3\rangle &:= \mathrm{CE}_V \, |\Psi_2\rangle \\ &= \frac{1}{\sqrt{N_c}} \sum_{j\in\{0,\,1\}^n} b_j^{\Gamma} \sum_{k\in\{0,\,1\}^{n_c}} e^{i2\pi\phi_j k_{(10)}} \, |\phi_j^{\Gamma}\rangle \, |k\rangle_c \, |0\rangle_a \,. \end{aligned} \tag{38}$$

Because $|\phi_j^{\Gamma}\rangle$ does not depend on k, it is convenient to move the accumulated phase to the clock register, leaving the input register $|\phi_j^{\Gamma}\rangle$ unchanged, namely

$$|\Psi_3\rangle = \left(\sum_{j\in\{0,1\}^n} b_j^{\Gamma} \, |\phi_j^{\Gamma}\rangle \otimes |\chi_j\rangle_c \right) \otimes |0\rangle_a \,, \tag{39}$$

where $|\chi_j\rangle_c$ is defined as

$$|\chi_j\rangle_c = \frac{1}{\sqrt{N_c}} \sum_{k\in\{0,1\}^{n_c}} e^{i2\pi\phi_j k_{(10)}} |k\rangle_c \,. \tag{40}$$

This phenomenon is general and it is called the kickback effect, which modifies the control state but leaves the target state unchanged. In other words, after applying the operator E_V to an eigenvector, the phase accumulates on the clock register. The phase kickback is a fundamental quantum phenomenon in which a phase acquired by a target quantum state during a controlled operation is effectively transferred back onto the control qubit. A simple example is reported in Appendix B. As a final remark of this step, it is important to note that the state given by Eq. (39) entangles the clock register with the input register. Because the phases ϕ_j make the control states $|\chi_j\rangle_c$ different for different j, the overall state becomes a sum of non-parallel product terms, which cannot be factored into a single tensor product – hence it is generically entangled. Now the clock register stores information about quantum phases ϕ_j, which we need to extract in the following step.

In the following step, let us apply the Quantum Fourier Transformation (QFT) [14] to the clock register of the quantum state given by Eq. (39), which means to apply QFT to $|\chi_j\rangle_c$ given by Eq. (40). The QFT is defined in this document in such a way so as to realize a unitary transformation by construction [26]. In particular, the generic component of $\hat{U}_{\text{QFT}}$ at the m-th row and at the l-th column is given by $1/\sqrt{N_c}\,\omega_N^{ml}$, namely

$$\hat{U}_{\text{QFT}} := \frac{1}{\sqrt{N_c}} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega_{N_c} & \dots & \omega_{N_c}^{N_c-1} \\ 1 & \omega_{N_c}^2 & \dots & \omega_{N_c}^{2(N_c-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{N_c}^{N_c-1} & \dots & \omega_{N_c}^{(N_c-1)(N_c-1)} \end{bmatrix}, \tag{41}$$

where $\omega_{N_c} = e^{i\,2\pi/N_c}$. $\hat{U}_{\text{QFT}}$ has dimensions $N_c \times N_c$, where $N_c = 2^{n_c}$ is the number of states of the clock register. This means that $\hat{U}_{\text{QFT}}^\dagger \hat{U}_{\text{QFT}} = \hat{U}_{\text{QFT}} \hat{U}_{\text{QFT}}^\dagger = I$, where $\hat{U}_{\text{QFT}}^\dagger$ denotes the conjugate transpose of $\hat{U}_{\text{QFT}}$ (namely Hermitian transpose). The quantum state of this step can be formally defined as

$$|\Psi_4\rangle := \left(\sum_{j\in\{0,1\}^n} b_j^\Gamma |\phi_j^\Gamma\rangle \otimes \hat{U}_{\text{QFT}}^\dagger |\chi_j\rangle_c \right) \otimes |0\rangle_a \,. \tag{42}$$

Let us elaborate on the previous expression by applying $\hat{U}_{\text{QFT}}^\dagger$ to $|\chi_j\rangle_c$ given by Eq. (40), namely

$$\begin{aligned} \hat{U}_{\text{QFT}}^\dagger |\chi_j\rangle_c &= \frac{1}{\sqrt{N_c}} \hat{U}_{\text{QFT}}^\dagger \sum_{k\in\{0,1\}^{n_c}} e^{i2\pi\phi_j k_{(10)}} |k\rangle_c \\ &= \frac{1}{\sqrt{N_c}} \sum_{k\in\{0,1\}^{n_c}} e^{i2\pi\phi_j k_{(10)}} \hat{U}_{\text{QFT}}^\dagger |k\rangle_c \,. \end{aligned} \tag{43}$$

Applying $\hat{U}_{\text{QFT}}^\dagger$ to a basis vector $|k\rangle_c$ yields the k-th column of the matrix $\hat{U}_{\text{QFT}}^\dagger$[1]

$$\hat{U}_{\text{QFT}}^\dagger |k\rangle_c = \frac{1}{\sqrt{N_c}} \sum_{\tilde{k}\in\{0,1\}^{n_c}} \omega_{N_c}^{-k_{(10)}\tilde{k}_{(10)}} |\tilde{k}\rangle_c \,. \tag{44}$$

Consequently

$$\begin{aligned} &\hat{U}_{\text{QFT}}^\dagger |\chi_j\rangle_c \\ &= \frac{1}{N_c} \sum_{k\in\{0,1\}^{n_c}} \sum_{\tilde{k}\in\{0,1\}^{n_c}} e^{i2\pi k_{(10)}\left[\phi_j - \tilde{k}_{(10)}/N_c\right]} |\tilde{k}\rangle_c \,. \end{aligned} \tag{45}$$

Using the last expression into Eq. (42) and swapping k with $\tilde{k}$ yield

$$\begin{aligned} |\Psi_4\rangle = &\sum_{j\in\{0,1\}^n} b_j^\Gamma |\phi_j^\Gamma\rangle \\ &\otimes \left(\frac{1}{N_c} \sum_{k\in\{0,1\}^{n_c}} \sum_{\tilde{k}\in\{0,1\}^{n_c}} e^{i2\pi\tilde{k}_{(10)}[\phi_j - k_{(10)}/N_c]} |k\rangle_c \right) \\ &\otimes |0\rangle_a \,. \end{aligned} \tag{46}$$

The last expression can be reformulated as

$$|\Psi_4\rangle = \sum_{j\in\{0,1\}^n} \left[b_j^\Gamma |\phi_j^\Gamma\rangle \otimes \left(\sum_{k\in\{0,1\}^{n_c}} \alpha_{jk}^{\text{QPE}} |k\rangle_c \right) \right] \otimes |0\rangle_a \,, \tag{47}$$

where the complex amplitude matrix α_{jk}^{QPE} is defined as

$$\begin{aligned} \alpha_{jk}^{\text{QPE}} &:= \frac{1}{N_c} \sum_{\tilde{k}\in\{0,1\}^{n_c}} e^{i2\pi\tilde{k}_{(10)}[\phi_j - k_{(10)}/N_c]} \\ &= \frac{1}{N_c} \sum_{\tilde{k}\in\{0,1\}^{n_c}} e^{i2\pi\tilde{k}_{(10)}\left[\lambda_j^\Gamma(N_c-1) - k_{(10)}\right]/N_c} , \end{aligned} \tag{48}$$

which used the definition given by Eq. (32). The last definition can be used to compact Eq. (45), namely

$$\hat{U}_{\text{QFT}}^\dagger |\chi_j\rangle_c = \sum_{k\in\{0,1\}^{n_c}} \alpha_{jk}^{\text{QPE}} |k\rangle_c \,. \tag{49}$$

Since $|\chi_j\rangle_c$ is clearly normalized and the operator $\hat{U}_{\text{QFT}}^\dagger$ is unitary, then

$$\sum_{k\in\{0,1\}^{n_c}} \left| \alpha_{jk}^{\text{QPE}} \right|^2 = 1, \tag{50}$$

which will be useful in the following. It is worth better understanding the term $\overline{m}_{jk} = \lambda_j^\Gamma(N_c - 1) - k_{(10)}$ in the complex amplitude matrix. Eq. (18) ensures that $0 < \lambda_j^\Gamma \leq$

[1] Let us consider a matrix A and a vector $\vec{a}$. Let us do the usual matrix-vector multiplication, namely $A \cdot \vec{a}$, which gives a vector. The generic component of the latter vector is $(A \cdot \vec{a})_\alpha$ and it can be expressed as $\sum_\beta A_{\alpha\beta}\, a_\beta$. Using the unit vector $\vec{e}_\alpha$, it is possible to express $A \cdot \vec{a} = \sum_\alpha \sum_\beta A_{\alpha\beta}\, a_\beta\, \vec{e}_\alpha$ or equivalently $A \cdot \vec{a} = \sum_\alpha c_\alpha \vec{e}_\alpha$ where $c_\alpha = \sum_\beta A_{\alpha\beta}\, a_\beta$. In case $\vec{a}$ is another unit vector $\vec{a}' = \vec{e}_\gamma$, then $c'_\alpha = \sum_\beta A_{\alpha\beta}\, \delta_{\gamma\beta} = A_{\alpha\gamma}$, which means that $\vec{c}\,'$ is the γ-th column of the matrix.

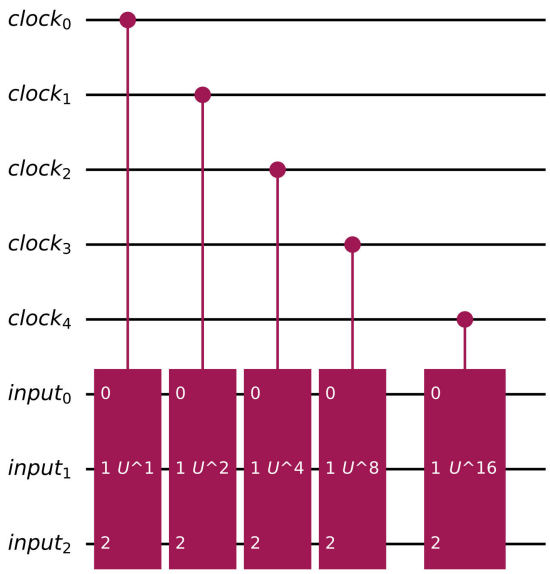

FIGURE 2. Circuit implementing controlled exponentiation (or controlled Hamiltonian evolution) in the HHL algorithm with $n = 3$ input qubits for solving the same problem discussed in the previous sections and $n_c = 5$ clock qubits. This circuit shows what is inside the block called "C-exp" in Fig. 1.

1 for all eigenvalues. Consequently $0 < (N_c - 1)\,\lambda_j^{\Gamma} \leq (N_c - 1)$, which means that $(N_c - 1)\,\lambda_j^{\Gamma}$ is included in the same numerical interval spanned by all the states of the clock register with n_c qubits. In other words, given $k \in \{0,\ 1\}^{n_c}$, then $k_{(10)} \in [0,\ N_c - 1]$ and also $(N_c - 1)\,\lambda_j^{\Gamma} \in [0,\ N_c - 1]$. The difference is that $k_{(10)}$ corresponds to the binary integer k, while $(N_c - 1)\,\lambda_j^{\Gamma}$ has no match with an integer in general. Consequently $\overline{m}_{jk}$ may be close to an integer, but it is not necessarily an integer. Eq. (48) can be simplified by realizing that it involves a finite geometric series, namely

$$\alpha_{jk}^{\mathrm{QPE}} = e^{if_{jk}} \frac{\sin(\pi\,\overline{m}_{jk})}{N_c\,\sin(\pi\,\overline{m}_{jk}/N_c)}, \tag{51}$$

where $f_{jk} = \pi(N_c - 1)\,\overline{m}_{jk}/N_c$. Before moving to the following module, it is interesting to realize that the quantum state given by Eq. (47) is entangled as well, because the clock register depends on the input register by the complex matrix $\alpha_{jk}^{\mathrm{QPE}}$.

D. EIGENVALUE INVERSION MODULE

The second essential component of the HHL algorithm is the eigenvalue inversion module. From a complexity-theoretic perspective, this module is crucial because it involves eigenvalue-dependent controlled rotations that are not restricted to the Clifford group [26]. The Clifford group is the set of quantum gates that map Pauli operators to Pauli operators under conjugation, and it can be efficiently simulated on a classical computer. In view of the Gottesman-Knill theorem [28], this non-Clifford structure of the eigenvalue inversion module is essential to prevent efficient classical simulation and thus underpins the potential quantum speedup of HHL [26]. There are many advanced and efficient quantum circuits for inverting eigenvalues, e.g. see [19]. Here we focus on a very simple straightforward implementation.

Before defining the operator for the inversion, let us estimate the minimum binarized eigenvalue which we have to invert. The binary clock register $k_{(10)}$ that we have to invert is very close to $\lambda_j^{\Gamma}(N_c - 1)$. Taking into account Eq. (18) yields

$$\frac{N_c - 1}{1 + 4\,r} \leq \lambda_j^{\Gamma}\,(N_c - 1) \leq (N_c - 1). \tag{52}$$

In order to take into account the round-off error and the measurement probability scattering, we can assume that the minimum binarized eigenvalue to be inverted can be expressed as

$$k_{(10)}^{\min} = \left\lfloor \frac{N_c - 1}{1 + 4\,r} \right\rfloor - \Delta N_{\epsilon}, \tag{53}$$

where ΔN_{ϵ} is a tolerance integer (e.g. in Fig. 3 we assumed $\Delta N_{\epsilon} = 2$ over $N_c = 2^5 = 32$). Knowing that $(k_j^{\lambda})_{(10)} > k_{(10)}^{\min}$ allows us to save some operations and to increase the sampling rate (as it will be clarified in the following). Therefore let us define the controlled rotation operator for inverting the eigenvalues as

$$\begin{aligned} &\mathrm{CR}_Y \\ &= I \otimes \left\{ \sum_{k \in \{0,\ 1\}^{n_c} < k^{\min}} |k\rangle_c\,\langle k|_c \otimes I_a \right. \\ &\quad \left. + \sum_{k \in \{0,\ 1\}^{n_c} \geq k^{\min}} |k\rangle_c\,\langle k|_c \otimes R_Y\left[2\arcsin(k_{(10)}^{\min}/k_{(10)})\right] \right\}. \end{aligned} \tag{54}$$

This operator is shown in Fig. 3 for the case with $n_c = 5$ clock qubits (plus the ancilla qubit). As it is clear from the circuit in Fig. 3, in this case $k_{(10)}^{\min} = 8$, which corresponds to $k^{\min} = 1000$ and is the controller condition of the first rotation block on the left. Starting with $k_{(10)}^{\min} = 8$ allows us to save seven controlled rotations ($k_{(10)} > 0$ for avoiding division by zero). On top of the saving, it is important to understand how this operator works. The term $\sum_{k \geq k_{\min}} |k\rangle_c\,\langle k|_c$ in Eq. (54) acts as a projection operator onto the subspace of clock-register states with $k \geq k^{\min}$, thereby enabling the appropriate controlled rotation associated with the inversion when a specific state $|k^*\rangle_c$ is encoded in the clock register. If the state $|k^*\rangle_c$ happens, namely

$$\left(\sum_{k \geq k_{\min}} |k\rangle_c\,\langle k|_c \right) |k^*\rangle_c = \sum_{k \geq k_{\min}} |k\rangle_c\,\delta_{kk^*} = |k^*\rangle_c\,, \tag{55}$$

then the ancilla qubit is rotated by an angle which is

$$\theta_Y^* = 2\arcsin\left[\frac{k_{(10)}^{\min}}{k_{(10)}^*}\right] = 2\arcsin\left(\alpha_k^*\right), \tag{56}$$

where $\alpha_k^* = k_{(10)}^{\min}/k_{(10)}^* < 1$. Clearly if $k^* = k^{\min}$, then $\theta_Y^* = \pi$, which is the leftmost rotation in Fig. 3. Taking into account the definition given by Eq. (86), applying the operator CR_Y to the previous system quantum state yields

$$\begin{aligned}|\Psi_5\rangle &:= \mathrm{CR}_Y\,|\Psi_4\rangle \\ &= \sum_{j\in\{0,\,1\}^n} b_j^\Gamma\,|\phi_j^\Gamma\rangle \otimes \Bigg[\sum_{k\in\{0,\,1\}^{n_c}<k^{\min}} \alpha_{jk}^{\mathrm{QPE}}\,|k\rangle_c \otimes |0\rangle_a \\ &\quad + \sum_{k\in\{0,\,1\}^{n_c}\geq k^{\min}} \alpha_{jk}^{\mathrm{QPE}}\,|k\rangle_c \otimes \left(\sqrt{1-\alpha_k^2}\;|0\rangle_a + \alpha_k\;|1\rangle_a\right)\Bigg].\end{aligned} \tag{57}$$

In the previous formula, the inverse eigenvalues implicitly appear but we need the following step for isolating this result.

The key idea here is to measure the ancilla qubit, to discard the calculation if the result is $|0\rangle$ and to keep it for statistics if it is $|1\rangle$. Following the Born rule, quantum mechanics tells us that, after a measurement, the post-measurement state $|\Psi_6\rangle$ must be normalized by dividing by the square root of the probability of the observed outcome, namely

$$\begin{aligned}|\Psi_6\rangle &:= \frac{1}{\sqrt{N_M}} \sum_{j\in\{0,\,1\}^n} b_j^\Gamma\;|\phi_j^\Gamma\rangle \\ &\quad \otimes \sum_{k\in\{0,\,1\}^{n_c}\geq k^{\min}} \left(\alpha_{jk}^{\mathrm{QPE}}\alpha_k\;|k\rangle_c\right) \otimes |1\rangle_a\,.\end{aligned} \tag{58}$$

It is interesting to note that the measurement of the ancilla transferred the inverse of the eigenvalues from the ancilla qubit to the clock register. The normalization factor N_M in the previous formula is required to ensure that $|\Psi_6\rangle$ is properly normalized (see next).

The previous state seems close to solving the problem, but the clock register is still entangled with the input register. Moreover the inverse eigenvalues in the clock register are unknown. The inverse of the operation that originally entangled them is required at this point, as discussed in the following section.

E. INVERSE QPE

As it is clear from the previous steps, the clock register has become an eigenvalue register. To obtain the "solution" in the input register, one must quantum-erase (uncompute) the eigenvalue register, i.e., apply exactly the inverse of the operation that originally encoded it, namely the QPE.

First of all, let us collect in one single unitary operation $\hat{U}_{\mathrm{QPE}}$ some previous steps, i.e. Hadamard transformation, Hamiltonian evolution and inverse QFT, namely

$$\hat{U}_{\mathrm{QPE}} = \left(I \otimes \hat{U}_{\mathrm{QFT}}^\dagger \otimes I_a\right)\,\mathrm{CE}_V\,\left(I \otimes H^{\otimes n_c} \otimes I_a\right). \tag{59}$$

The logical order goes from right to left: first, $\left(I \otimes H^{\otimes n_c} \otimes I_a\right)|\Psi_1\rangle = |\Psi_2\rangle$; then, $\mathrm{CE}_V\,|\Psi_2\rangle = |\Psi_3\rangle$ and finally, $\left(I \otimes \hat{U}_{\mathrm{QFT}}^\dagger \otimes I_a\right)|\Psi_3\rangle = |\Psi_4\rangle$. Using the newly defined operator, these three steps, which are essential for the QPE algorithm, can be summarized as $\hat{U}_{\mathrm{QPE}}\,|\Psi_1\rangle = |\Psi_4\rangle$. The eigenvalue inversion module allowed to compute $|\Psi_6\rangle$. Hence the already-mentioned quantum uncomputation can be formally realized by the inverse operator $\hat{U}_{\mathrm{QPE}}^\dagger$, namely

$$\begin{aligned}|\Psi_9\rangle &:= \hat{U}_{\mathrm{QPE}}^\dagger\,|\Psi_6\rangle = \frac{1}{\sqrt{N_M}} \sum_{j\in\{0,\,1\}^n} b_j^\Gamma \\ &\quad \cdot \sum_{k\in\{0,\,1\}^{n_c}\geq k^{\min}} \alpha_{jk}^{\mathrm{QPE}}\alpha_k\;\hat{U}_{\mathrm{QPE}}^\dagger\,|\phi_j^\Gamma\rangle\,|k\rangle_c\,|1\rangle_a\,.\end{aligned} \tag{60}$$

Although some simplifications are possible at the cost of accuracy, the optimal HHL algorithm requires an additional projection onto the ground state of the clock register, i.e. $|0\rangle_c^{\otimes n_c}$, which can be implemented through an appropriate measurement. For the generic j-th mesh node the probability of the corresponding state can be computed by the following projection, which can be conveniently written as

$$a_j = b_j^\Gamma\;g(\lambda_j^\Gamma) = \langle\phi_j^\Gamma|\,\langle 0|_c^{\otimes n_c}\,\langle 1|_a\,|\Psi_9\rangle\,, \tag{61}$$

where $g(\lambda_j^\Gamma)$ is sometimes called "filter" and is a proper function which will be clarified in the following. Equivalently, we can express it as

$$\begin{aligned}a_j = b_j^\Gamma\;g(\lambda_j^\Gamma) &= \sum_{j'\in\{0,\,1\}^n} b_{j'}^\Gamma \sum_{k\in\{0,\,1\}^{n_c}\geq k^{\min}} \alpha_{j'k}^{\mathrm{QPE}}\alpha_k \\ &\quad \cdot \langle\phi_j^\Gamma|\,\langle 0|_c^{\otimes n_c}\,\langle 1|_a\,\hat{U}_{\mathrm{QPE}}^\dagger\,|\phi_{j'}^\Gamma\rangle\,|k\rangle_c\,|1\rangle_a\,.\end{aligned} \tag{62}$$

Please note that, in order to avoid confusion with j of the amplitude a_j we are searching for, we introduced the index j' in the summation. In order to compute the quantum measurement in the previous expression, let us take into account the following property

$$\begin{aligned}&\langle\phi_j^\Gamma|\,\langle 0|_c^{\otimes n_c}\,\langle 1|_a\,\hat{U}_{\mathrm{QPE}}^\dagger\,|\phi_{j'}^\Gamma\rangle\,|k\rangle_c\,|1\rangle_a \\ &= \left(\langle\phi_{j'}^\Gamma|\,\langle k|_c\,\langle 1|_a\,\hat{U}_{\mathrm{QPE}}\,|\phi_j^\Gamma\rangle\,|0\rangle_c^{\otimes n_c}\,|1\rangle_a\right)^*,\end{aligned} \tag{63}$$

where the superscript * indicates the complex conjugate, as usual. The result $\hat{U}_{\mathrm{QPE}}\,|\phi_j^\Gamma\rangle\,|0\rangle_c^{\otimes n_c}\,|1\rangle_a$ can be computed as a particular case of Eq. (60), i.e. assuming $|\Psi_9\rangle = |\phi_j^\Gamma\rangle$, namely

$$\begin{aligned}&\hat{U}_{\mathrm{QPE}}\,|\phi_j^\Gamma\rangle\,|0\rangle_c^{\otimes n_c}\,|1\rangle_a \\ &= |\phi_j^\Gamma\rangle \otimes \sum_{k'\in\{0,\,1\}^{n_c}} \left(\alpha_{jk'}^{\mathrm{QPE}}\;|k'\rangle_c\right) \otimes |1\rangle_a\,,\end{aligned} \tag{64}$$

where again index k' is introduced for avoiding confusion. Applying the result given by Eq. (64) into Eq. (63) yields

$$\langle\phi_j^\Gamma|\,\langle 0|_c^{\otimes n_c}\,\langle 1|_a\,\hat{U}_{\mathrm{QPE}}^\dagger\,|\phi_{j'}^\Gamma\rangle\,|k\rangle_c\,|1\rangle_a = \delta_{j'j}\left(\alpha_{jk}^{\mathrm{QPE}}\right)^*. \tag{65}$$

Applying the last expression into Eq. (62) yields

$$\begin{aligned}a_j &= b_j^\Gamma\;g(\lambda_j^\Gamma) \\ &= \sum_{j'\in\{0,\,1\}^n} b_{j'}^\Gamma \sum_{k\in\{0,\,1\}^{n_c}\geq k^{\min}} \alpha_{j'k}^{\mathrm{QPE}}\alpha_k\,\delta_{j'j}\left(\alpha_{jk}^{\mathrm{QPE}}\right)^*,\end{aligned} \tag{66}$$

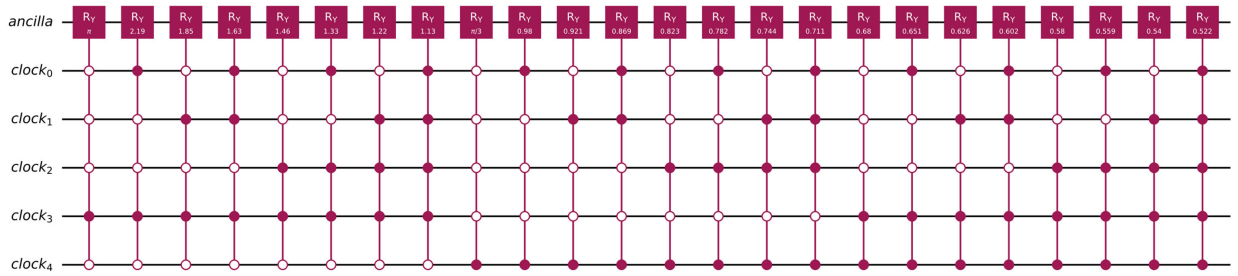

FIGURE 3. Circuit implementing controlled rotation on the ancilla qubit in the HHL algorithm with $n_c = 5$ clock qubits. As it is clear from the circuit, $k_{(10)}^{\min} = 8$, which corresponds to $k^{\min} = 1000$ and is the controller condition of the first rotation block on the left. This circuit shows what is inside the block called "CRy-for-inversion" in Fig. 1.

or equivalently

$$g(\lambda_j^\Gamma) = a_j / b_j^\Gamma = \sum_{k \in \{0,\,1\}^{n_c} \geq k^{\min}} w_{jk}\, \alpha_k, \tag{67}$$

where

$$w_{jk} = \left|\alpha_{jk}^{\mathrm{QPE}}\right|^2 = \left|\frac{\sin(\pi\, \overline{m}_{jk})}{N_c\, \sin(\pi\, \overline{m}_{jk}/N_c)}\right|^2. \tag{68}$$

Taking into account Eq. (50), it is possible to realize that the previous terms are actually normalized weights, namely

$$\sum_{k \in \{0,\,1\}^{n_c}} w_{jk} = 1, \tag{69}$$

for every j-th component. See also Fig. 4 for an example. After computing the amplitudes $a_j = b_j^\Gamma\, g(\lambda_j^\Gamma)$, we can use them to normalize the final state as

$$|\Psi_9\rangle = \frac{1}{\sqrt{\sum_j \left|b_j^\Gamma\, g(\lambda_j^\Gamma)\right|^2}} \cdot \sum_{j \in \{0,\,1\}^n} b_j^\Gamma\, g(\lambda_j^\Gamma)\, |\phi_j^\Gamma\rangle\, |0\rangle_c^{\otimes n_c}\, |1\rangle_a\,. \tag{70}$$

In the previous final state, for the generic j-th component of the input decomposition, we can define the actual HHL eigenvalue as

$$\begin{aligned} \frac{1}{\lambda_j^{\mathrm{HHL}}} &:= \frac{(N_c - 1)}{k_{(10)}^{\min}}\, g(\lambda_j^\Gamma) \\ &= \frac{(N_c - 1)}{k_{(10)}^{\min}} \sum_{k \in \{0,\,1\}^{n_c} \geq k^{\min}} w_{jk}\, \alpha_k \\ &= \sum_{k \in \{0,\,1\}^{n_c} \geq k^{\min}} w_{jk}\, \frac{(N_c - 1)}{k_{(10)}}. \end{aligned} \tag{71}$$

The HHL eigenvalues can be used to define the coefficients of the decomposition of the HHL solution, which appear in the pre-factors of the final quantum state, namely

$$\frac{b_j^\Gamma\, g(\lambda_j^\Gamma)}{\sqrt{\sum_j \left|b_j^\Gamma\, g(\lambda_j^\Gamma)\right|^2}} = \frac{b_j^\Gamma / \lambda_j^{\mathrm{HHL}}}{\sqrt{\sum_j \left|b_j^\Gamma / \lambda_j^{\mathrm{HHL}}\right|^2}} = x_j^{\mathrm{HHL}}. \tag{72}$$

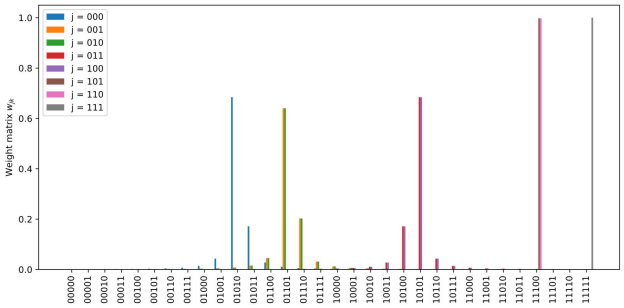

FIGURE 4. Normalized weights w_{jk}, where $\sum_k w_{jk} = 1$ for every j-th component of the spectral decomposition. It is worth noting that the binarization process implies $(N_c - 1)\,\lambda_j^\Gamma \approx (k_j^\lambda)_{(10)}$, which introduces some spread in the spectral distribution of the binarized eigenvalues. For example, $(N_c - 1)\,\lambda_0^\Gamma = 10.33$, which falls in between $k = 01010$ and $k' = 01011$.

Introducing these coefficients in Eq. (70) yields

$$\begin{aligned} |\Psi_9\rangle &= \sum_{j \in \{0,\,1\}^n} x_j^{\mathrm{HHL}}\, |\phi_j^\Gamma\rangle\, |0\rangle_c^{\otimes n_c}\, |1\rangle_a \\ &= |x^{\mathrm{HHL}}\rangle\, |0\rangle_c^{\otimes n_c}\, |1\rangle_a\,, \end{aligned} \tag{73}$$

where the state $|x^{\mathrm{HHL}}\rangle$ is the quantum state produced by the HHL algorithm. The latter quantum state, employed to compute the single time-step update of the temperature profile according to the heat conduction equation using the HHL algorithm (3 qubits), is shown in Fig. 5.

In the next section, we will use a simplified derivation of the HHL algorithm to show that $\lambda_j^{\mathrm{HHL}} \approx \lambda_j^\Gamma$ and consequently that $|x^{\mathrm{HHL}}\rangle \approx |x\rangle$.

IV. SIMPLIFIED ANALYSIS OF THE HHL ALGORITHM

In the previous sections, we have already pointed out that the term $\overline{m}_{jk} = \lambda_j^\Gamma\,(N_c - 1) - k_{(10)}$ is not necessarily an integer in general. Eq. (18) ensures that $0 < \lambda_j^\Gamma \leq 1$ for all eigenvalues. Consequently $0 < (N_c - 1)\,\lambda_j^\Gamma \leq (N_c - 1)$, which means that $(N_c - 1)\,\lambda_j^\Gamma$ is included in the same numerical interval $[0,\, N_c - 1]$, as it happens for $k_{(10)}$. The difference is that $k_{(10)}$ corresponds to the binary integer k, while $(N_c - 1)\,\lambda_j^\Gamma$ has no

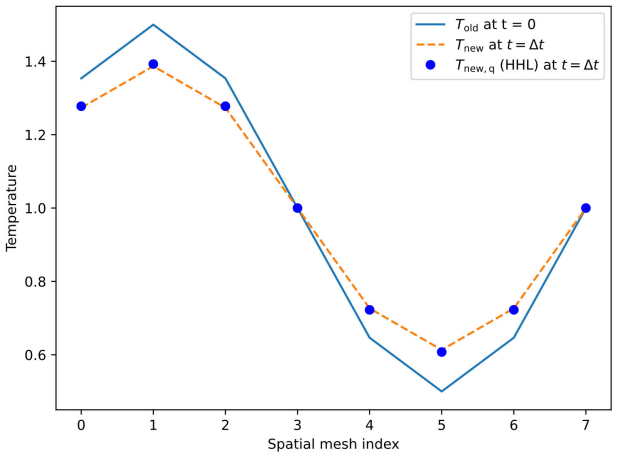

FIGURE 5. One time-step update of the temperature profile according to heat conduction equation by quantum computing (3 qubits) by HHL algorithm. The blue line is the initial temperature profile (with mean equal to 1), the orange dashed line is the new temperature profile at time Δt, computed by finite-difference method. The blue dots are the mesh node temperatures computed by the HHL algorithm (ideal statevector simulator). The circuit implementing HHL algorithm with $n = 3$ input qubits, $n_c = 5$ clock qubits and 1 ancilla qubit is shown in Fig. 1. In particular, the eigenvalue inversion module with $n_c = 5$ qubits in the clock register is shown in Fig. 3.

match with an integer in general. Let us relax here the last condition, namely

$$(N_c - 1)\,\lambda_j^{\Gamma} \approx (k_j^{\lambda})_{(10)}, \tag{74}$$

where $k_j^{\lambda} \in \mathbb{Z}$ is a proper integer. In this case, the term $\overline{m}_{jk} \approx (k_j^{\lambda})_{(10)} - k_{(10)} = m_{jk} \in \mathbb{Z}$ and consequently $\alpha_{jk}^{\text{QPE}} \approx \delta_{k\,k_j^{\lambda}}$, or equivalently $w_{jk} \approx \delta_{k\,k_j^{\lambda}}$. It is interesting to see the impact of the approximation given by Eq. (74) in the derivation of the previous sections. In particular, substituting it in Eq. (45) yields

$$\begin{aligned}&\hat{U}_{\text{QFT}}^{\dagger}\,|\chi_j\rangle_c \\ &\approx \frac{1}{N_c}\sum_{k\in\{0,\,1\}^{n_c}}\sum_{\tilde{k}\in\{0,\,1\}^{n_c}} e^{i2\pi k_{(10)}\left(k_j^{\lambda}-\tilde{k}\right)_{(10)}/N_c}\,|\tilde{k}\rangle_c\,.\end{aligned} \tag{75}$$

Because now k_j^{λ} are binary integers, only the term with $\tilde{k} = k_j^{\lambda}$ is non-zero, which simplifies the previous expression as

$$\hat{U}_{\text{QFT}}^{\dagger}\,|\chi_j\rangle_c \approx \frac{1}{N_c}\sum_{k\in\{0,\,1\}^{n_c}} e^{0}\,|k_j^{\lambda}\rangle_c = |k_j^{\lambda}\rangle_c\,, \tag{76}$$

which is perfectly consistent with $\alpha_{jk}^{\text{QPE}} \approx \delta_{k\,k_j^{\lambda}}$ and Eq. (49). Applying the last result to the generic quantum state given by Eq. (42) yields

$$|\Psi_4\rangle \approx |\Psi_4'\rangle := \sum_{j\in\{0,\,1\}^n}\left(b_j^{\Gamma}\,|\phi_j^{\Gamma}\rangle \otimes |k_j^{\lambda}\rangle_c\right)\otimes|0\rangle_a\,. \tag{77}$$

In ideal (statevector) simulations (otherwise, the wave function would collapse in real computations), it is possible to compute the probability distribution of the binarized approximations of the eigenvalues of the problem $|k_j^{\lambda}\rangle$ stored in the clock register. See an example in Fig. 6. It is worth noting that the binarization process implies $(k_j^{\lambda})_{(10)} \approx (N_c - 1)\,\lambda_j^{\Gamma}$, which introduces some spread in the spectral distribution of the binarized eigenvalues (see Fig. 4). However, in some cases, this spread may even be beneficial for the accuracy of the numerical solution produced by the HHL algorithm.

Proceeding as described in the previous sections for inverting the binarized eigenvalues and taking advantage of the assumption given by $\alpha_{jk}^{\text{QPE}} \approx \delta_{k\,k_j^{\lambda}}$ in Eq. (58) yield

$$|\Psi_6\rangle \approx |\Psi_6'\rangle := \frac{1}{\sqrt{N_M}}\sum_{j\in\{0,\,1\}^n}\alpha_j b_j^{\Gamma}\,|\phi_j^{\Gamma}\rangle\,|k_j^{\lambda}\rangle_c\,|1\rangle_a\,, \tag{78}$$

where $\alpha_j = k_{(10)}^{\min}/(k_j^{\lambda})_{(10)}$. In order to derive N_M, it is possible to focus on the input register and the clock register only. Using (i) the tensor-product property on inner products and (ii) the orthogonality of the eigenvectors yields

$$\begin{aligned}N_M &= \sum_{j\in\{0,\,1\}^n}\alpha_j\,b_j^{\Gamma}\,\langle\phi_j^{\Gamma}|\,\langle k_j^{\lambda}|_c\sum_{j'\in\{0,\,1\}^n}\alpha_{j'}\,b_{j'}^{\Gamma}\,|\phi_{j'}^{\Gamma}\rangle\,|k_{j'}^{\lambda}\rangle_c \\ &= \sum_{j,j'\in\{0,\,1\}^n}\alpha_j\,b_j^{\Gamma}\,\alpha_{j'}\,b_{j'}^{\Gamma}\,\langle\phi_j^{\Gamma}|\,\langle k_j^{\lambda}|_c\,|\phi_{j'}^{\Gamma}\rangle\,|k_{j'}^{\lambda}\rangle_c \\ &= \sum_{j,j'\in\{0,\,1\}^n}\alpha_j\,b_j^{\Gamma}\,\alpha_{j'}\,b_{j'}^{\Gamma}\,\langle\phi_j^{\Gamma}|\phi_{j'}^{\Gamma}\rangle\,\langle k_j^{\lambda}|k_{j'}^{\lambda}\rangle \\ &= \sum_{j,j'\in\{0,\,1\}^n}\alpha_j\,b_j^{\Gamma}\,\alpha_{j'}\,b_{j'}^{\Gamma}\,\delta_{jj'}\,\langle k_j^{\lambda}|k_{j'}^{\lambda}\rangle \\ &= \sum_{j\in\{0,\,1\}^n}|\alpha_j\,b_j^{\Gamma}|^2\,\langle k_j^{\lambda}|k_j^{\lambda}\rangle \\ &= \sum_{j\in\{0,\,1\}^n}|\alpha_j\,b_j^{\Gamma}|^2.\end{aligned} \tag{79}$$

The previous expression completes the normalization factor appearing in Eq. (78). As discussed in the previous sections, we now need to project the state $|\Psi_6'\rangle$ onto the ground state of the clock register, namely

$$\begin{aligned}|\Psi_9\rangle \approx |\Psi_9'\rangle &:= \hat{U}_{\text{QPE}}^{\dagger}\,|\Psi_6'\rangle \\ &= \frac{1}{\sqrt{\sum_j\left|\alpha_j\,b_j^{\Gamma}\right|^2}}\sum_{j\in\{0,\,1\}^n}\alpha_j\,b_j^{\Gamma}\,|\phi_j^{\Gamma}\rangle\,|0\rangle_c^{\otimes n_c}\,|1\rangle_a\,.\end{aligned} \tag{80}$$

Comparing Eq. (70) and Eq. (80), it is clear that

$$g(\lambda_j^{\Gamma}) \approx \alpha_j = k_{(10)}^{\min}/(k_j^{\lambda})_{(10)}, \tag{81}$$

which is consistent with the assumption $w_{jk} \approx \delta_{k\,k_j^{\lambda}}$. In the previous final state, for the generic j-th component of the input decomposition, we can define the generic simplified eigenvalue as

$$\frac{1}{\lambda_j^{\text{SIM}}} := \frac{(N_c - 1)}{(k_j^{\lambda})_{(10)}}. \tag{82}$$

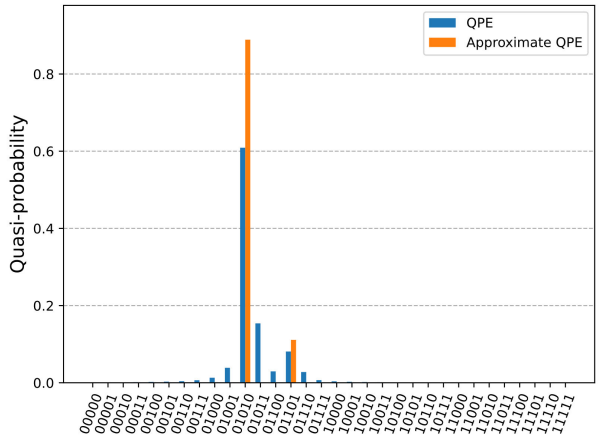

FIGURE 6. Probability distribution of the binarized approximations of the eigenvalues of the problem $|k_j^\lambda\rangle$ stored in the clock register, after completing QPE. Accessing this information is only possible in ideal (statevector) simulations (otherwise, the wave function would collapse in real computations). The two peaks correspond to (i) eigenvalue $k_0^\lambda = 01010$ or equivalently $(k_0^\lambda)_{(10)} = 10$, and to (ii) eigenvalue $k_1^\lambda = 01101$ or equivalently $(k_1^\lambda)_{(10)} = 13$. It is worth noting that the binarization process implies $(k_j^\lambda)_{(10)} \approx (N_c - 1)\lambda_j^\Gamma$, which introduces some spread in the spectral distribution of the binarized eigenvalues (see Fig. 4).

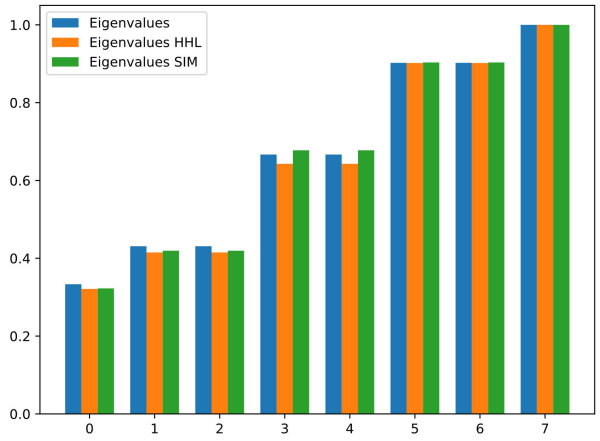

FIGURE 7. Comparison among eigenvalues: (i) original ones for the considered example; (ii) those approximated by the HHL algorithm; (iii) those computed by mimicking the HHL algorithm, but assuming the approximation given by Eq. (74). The abscissa represents the index labels ranging from 0 to $N-1$, corresponding to the normalized eigenvalues of the matrix $\hat{\Gamma}$ given by Eq. (16) in increasing order. By design, the maximum normalized eigenvalue is equal to one and it is exactly represented by the binary integer $\{1\}^{n_c}$. The approximated eigenvalues reported with label 5 and 6 differ from the exact values by less than 0.1 % and hence the discrepancies are not visible.

The simplified eigenvalues can be used in the coefficients of the decomposition of the solution, which appear in the pre-factors of the final quantum state, namely

$$\frac{\alpha_j\, b_j^\Gamma}{\sqrt{\sum_j \left|\alpha_j\, b_j^\Gamma\right|^2}} = \frac{b_j^\Gamma/(k_j^\lambda)_{(10)}}{\sqrt{\sum_j \left|b_j^\Gamma/(k_j^\lambda)_{(10)}\right|^2}}$$

$$= \frac{1}{\sqrt{\sum_j \left|b_j^\Gamma/\lambda_j^{\mathrm{SIM}}\right|^2}} \frac{b_j^\Gamma}{\lambda_j^{\mathrm{SIM}}} = x_j^{\mathrm{SIM}}. \quad (83)$$

Substituting the previous result in Eq. (80) yields

$$|\Psi_9'\rangle = \sum_{j\in\{0,\,1\}^n} x_j^{\mathrm{SIM}}\, |\phi_j^\Gamma\rangle\, |0\rangle_c^{\otimes n_c}\, |1\rangle_a$$
$$= |x^{\mathrm{SIM}}\rangle \otimes |0\rangle_c^{\otimes n_c} \otimes |1\rangle_a\,. \quad (84)$$

Substituting the fundamental hypothesis given by Eq. (74) into Eq. (82) yields

$$\frac{1}{\lambda_j^{\mathrm{SIM}}} = \frac{(N_c-1)}{(k_j^\lambda)_{(10)}} \approx \frac{1}{\lambda_j^\Gamma}. \quad (85)$$

Interestingly, the simplified derivation of the HHL algorithm discussed in this section also provides an intuitive explanation of why the algorithm yields a good approximation of the solution to the original problem. In fact, QPE makes it possible to extract binarized distributions of the original eigenvalues, as shown in Fig. 6. These distributions can then be used to compute an approximate estimate of the eigenvalues of the original problem, as illustrated in Fig. 7. Therefore, it becomes clear that $\lambda_j^{\mathrm{HHL}} \approx \lambda_j^{\mathrm{SIM}} \approx \lambda_j^\Gamma$. Consequently, the HHL algorithm provides an approximate solution to the original problem, namely $|x^{\mathrm{HHL}}\rangle \approx |x^{\mathrm{SIM}}\rangle \approx |x\rangle$, as shown in Fig. 5 for the considered case. As discussed in Appendix C, the HHL numerical solutions exhibit excellent agreement with the reference solution and achieve an accuracy comparable to that obtained with variational quantum algorithms [7].

However, there is an important difference between the HHL algorithm and the variational algorithms [7]. The key idea behind HHL is to retain only those quantum states for which the ancilla qubit is in the state $|1\rangle_a$ and, simultaneously, the clock register is in the ground state, i.e. $|0\rangle_c^{\otimes n_c}$. In the present example, the states satisfying $|0\rangle_c^{\otimes n_c}\, |1\rangle_a$ account for only 55.5 % of the total number of states. This implies that, statistically, about half of the measurement outcomes in actual shot-based simulations must be discarded when implementing the HHL algorithm. This is generally not considered a significant issue once quantum computers reach full maturity.

V. CONCLUSION

In this work, the HHL algorithm has been presented through a detailed analytical treatment, complemented by a complete example drawn from thermal engineering. This contribution is conceived as a follow-up tutorial to the previous tutorial by Zaman et al. [14], with the two works intentionally designed to be synergistic. While the previous tutorial establishes the fundamental physical and mathematical framework of the HHL algorithm, the present work extends that foundation toward more advanced and application-oriented aspects, maintaining consistency in notation and methodological approach. Together, the two tutorials aim to provide a coherent and progressively structured learning path.

More specifically, the present tutorial introduces two distinctive features that, to the best of our knowledge, are not simultaneously addressed in existing educational resources. First, it presents a systematic controlled-rotation procedure that does not rely on prior knowledge of the eigenvalues, thereby providing a more general framework than the simplified constructions often adopted in introductory presentations. Second, the derivation retains the non-integer binarized eigenvalue terms throughout the analysis in the complex amplitude matrix given by Eq. (48), allowing readers to follow the exact mathematical development before introducing the commonly used integer-eigenvalue approximation in a separate section.

The selected engineering example, previously investigated using a variational quantum approach for comparison [7], offers a concrete context in which the structure of HHL can be analyzed without restrictive simplifications. The accompanying Jupyter Notebook implementation [20] in Qiskit [16] further supports a practical understanding of the algorithmic workflow and its circuit-level realization.

More broadly, this tutorial has sought to deepen the conceptual understanding of quantum linear system algorithms during the current transitional phase of quantum hardware development. Although large-scale, noise-resilient quantum computers are still under development, rigorous and carefully structured tutorials remain essential for consolidating shared knowledge and extending it to the broader engineering community. In particular, a careful examination of the eigenvalue inversion module reveals that HHL does not compute matrix inverses in the classical sense; rather, it coherently encodes a discrete set of pre-defined inverse eigenvalue amplitudes and selectively weights them according to the eigenvalues of the specific problem under consideration. Such insights are crucial for developing realistic expectations about the capabilities and limitations of quantum algorithms.

At the same time, a realistic perspective must be maintained. While HHL remains a paradigmatic example of potential exponential quantum speedup under assumptions of sparsity, favorable conditioning, and efficient data representation, its implementation on current NISQ devices is severely constrained. The algorithm requires deep quantum circuits involving phase estimation, controlled rotations, and precise eigenvalue inversion—operations that are highly sensitive to noise and limited circuit depth. As a consequence, the practical deployment of HHL must await the availability of large-scale, fault-tolerant quantum computers.

In summary, HHL should presently be regarded both as a theoretically powerful algorithm and as a valuable pedagogical benchmark. By positioning this work as a follow-up to the previous tutorial in [14], and designing the two contributions to operate in synergy, we aim to support a structured and cumulative understanding of quantum linear system algorithms at the interface between quantum information science and engineering applications.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

Portions of this work were developed with the assistance of ChatGPT, an AI language model by OpenAI, in accordance with the CC-BY 4.0 license. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

APPENDIX

A. REAL DATA LOADING/ENCODING

Loading real-world input data into a quantum processor is a fundamental prerequisite for efficient quantum computation and requires the preparation of an appropriate quantum state. Most quantum computing software frameworks (e.g., [16], [18]) provide dedicated routines for state preparation. Several algorithms exploit the full set of quantum degrees of freedom to generate arbitrary quantum states while attempting to minimize the circuit depth. In the context of quantum circuits, the depth refers to the number of sequential (time-ordered) layers of quantum gates required to execute an algorithm, and thus quantifies the number of computational steps needed to process information. Divide-and-conquer strategies have been proposed to load an N-dimensional vector using quantum circuits with polylogarithmic depth [24]. However, these approaches require a number of classical input parameters that scales exponentially with the number of qubits, namely $\sim N = 2^n$ (up to normalization). Consequently, for very large problem sizes, the classical preprocessing and data transfer requirements become prohibitively demanding.

As an example, let us discuss here the divide-and-conquer paradigm [24]. The main idea is to divide a problem into subproblems of the same class and combine the solutions of the subproblems, in a recursive way, to obtain the solution of the original problem [24]. In particular, one of the standard methods for loading information in a quantum device is based on using controlled rotations [24]. In case of loading real values, it is enough to consider the rotation θ_Y around the Y-axis of the Bloch sphere given by the operator $R_Y(\theta_Y)$ [26], defined as

$$R_Y(\theta_Y) := \begin{bmatrix} \cos(\theta_Y/2) & -\sin(\theta_Y/2) \\ \sin(\theta_Y/2) & \cos(\theta_Y/2) \end{bmatrix}. \tag{86}$$

Readers who are not familiar with the Bloch sphere representation may consult the construction detailed in [7]. Clearly, $R_Y|0\rangle = a_0|0\rangle + a_1|1\rangle$, where a_0 and a_1 are real, in this case, amplitudes which satisfy the normalization condition $a_0^2 + a_1^2 = 1$. Inverting the previous relation for loading the desired amplitude yields $\theta_Y = 2\arcsin a_1$.

The divide-and-conquer paradigm relies on sequences of controlled rotations constructed according to an appropriate binary tree representation of the data to be loaded. Interested readers are referred to [24] for a detailed discussion. Here, we illustrate the procedure for the simple case $n = 2$, which can be derived explicitly in just a few steps. For $n = 2$, a generic quantum state can be written as $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$, with normalization condition

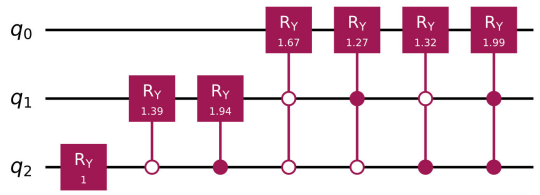

FIGURE 8. Quantum circuit for loading real data in a quantum system state $|x\rangle$ based on a divide-and-conquer algorithm proposed in [24]. Note that the proposed circuit uses only rotations about the Y-axis (R_Y gates), which are designed by means of a proper binary tree data structure [24]. For clarity, horizontal lines represent quantum wires which correspond to qubits in the circuit, red squares are the R_Y gates (see Eq. (86)), red dots represent control points in controlled gates and empty red circles represent anti-control points (i.e. activated by zero value). The little-endian convention is used [26], meaning that the topmost qubit represents the least significant bit (LSB), while the bottom qubit corresponds to the most significant bit (MSB), which, in this case, forms the trunk of the state tree in the divide-and-conquer strategy [24].

$a_{00}^2 + a_{01}^2 + a_{10}^2 + a_{11}^2 = 1$. Let p_{1q} denote the probability that the most significant qubit is in the state $|1\rangle$, independently of the state of the least significant qubit. This probability is $p_{1q} = a_{10}^2 + a_{11}^2$. Applying the rotation $R_Y(\theta_{yq})$ to the most significant qubit, with angle $\theta_{yq} = 2 \arcsin \sqrt{p_{1q}}$, ensures that the states $|10\rangle$ and $|11\rangle$ occur with total probability p_{1q}, while $|00\rangle$ and $|01\rangle$ occur with total probability $p_{0q} = 1 - p_{1q}$. Next, a controlled rotation $R_Y(\theta_{1y})$ is applied to the least significant qubit, conditioned on the most significant qubit being in state $|1\rangle$, where $\theta_{1y} = 2 \arcsin \sqrt{p_{11}/p_{1q}}$ and $p_{11} = a_{11}^2$. Similarly, a controlled rotation $R_Y(\theta_{0y})$ is applied to the least significant qubit, conditioned on the most significant qubit being in state $|0\rangle$, with $\theta_{0y} = 2 \arcsin \sqrt{p_{01}/p_{0q}}$ and $p_{01} = a_{01}^2$. Starting from the initial state $|00\rangle$, the sequence consisting of $R_Y(\theta_{yq})$, followed by the 1-controlled rotation $R_Y(\theta_{1y})$ and the 0-controlled rotation $R_Y(\theta_{0y})$, prepares the desired quantum state $|\psi\rangle$ at the end of the circuit execution.

Extending the previous approach to the case of $n = 3$ qubits–relevant for the example discussed in the main text–the quantum circuit $\hat{L}$ used to load the initial state $|b\rangle$ is shown in Fig. 8. The state $|b\rangle$ is obtained by normalizing the initial temperature profile $\vec{T} \equiv \vec{T}^{t=0}$ given in Eq. (9). The circuit $\hat{L}$ implements the divide-and-conquer strategy described above and enables the preparation of the desired probability amplitudes corresponding to the components of the normalized vector, namely

$$\hat{L} \, |0\rangle^{\otimes n} = |b\rangle = \sum_{j \in \{0,\,1\}^n} b_j \, |j\rangle \, . \tag{87}$$

B. PHASE KICKBACK

Phase kickback is a fundamental quantum phenomenon in which a phase acquired by a target quantum state during a controlled operation is effectively transferred back onto the control qubit [26]. When a controlled-unitary acts on an eigenstate of the unitary gate U, the eigenvalue's phase factor—normally applied to the target—appears instead as a relative phase on the control qubit, leaving the target unchanged. This surprising "kickback" of phase enables many key quantum algorithms, most notably Quantum Phase Estimation (QPE), where the eigenphase of a unitary is written onto the clock qubits, allowing extraction of otherwise inaccessible phase information using interference and measurement.

In order to understand phase kickback, let us consider first the Pauli-Y matrix [26] defined as

$$Y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} . \tag{88}$$

The eigenstates of the previous Pauli-Y matrix can be easily computed as

$$|y\pm\rangle = \frac{|0\rangle \pm i \, |1\rangle}{\sqrt{2}} , \tag{89}$$

which ensure by definition $Y \, |y\pm\rangle = \pm \, |y\pm\rangle$. Applying rotation $R_Y(\theta_Y)$ given by Eq. (86) to the previous states yields

$$R_Y(\theta_Y) \, |y\pm\rangle = e^{\mp \, i \, \theta_Y / 2} \, |y\pm\rangle \, . \tag{90}$$

This looks like an eigenvalue equation, but the eigenvalue is only a global phase, which has no physical effect because it cannot be measured. In particular, let us consider

$$|y-\rangle = \frac{|0\rangle - i \, |1\rangle}{\sqrt{2}} , \tag{91}$$

which can be prepared by applying a Hadamard gate H followed by the Hermitian adjoint of the phase gate $S^\dagger$, as shown for qubit q_1 in Fig. 9. The system quantum state at the first barrier of the circuit shown in Fig. 9 is

$$\begin{aligned} |\Psi_{\text{before}}\rangle &= |y-\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle - i \, |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2} \left(|00\rangle + |01\rangle - i \, |10\rangle - i \, |11\rangle \right) . \end{aligned} \tag{92}$$

This state can also be represented in column vector representation with regards to the computational basis $\{|00\rangle \, , |01\rangle \, , |10\rangle \, , |11\rangle\}$, namely

$$|\Psi_{\text{before}}\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -i \\ -i \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ -i \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} . \tag{93}$$

In this example, let us now suppose to consider the rotation $R_Y(\pi)$ as the unitary gate U_π, namely

$$U_\pi := R_Y(\pi) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} . \tag{94}$$

Let us now couple the two qubits of the example in Fig. 9. When composing physical systems, the sequential labeling of their components (e.g., $|\psi_0\rangle \, , |\psi_1\rangle \, , \ldots , |\psi_{n-1}\rangle$) may differ from the mathematical notation used to represent the bit strings, i.e., $\beta_{n-1} \ldots \beta_1 \beta_0$. In this case, in order to avoid confusion with the HHL algorithm in section III, let us align

the two notations, namely $|q_1 q_0\rangle$. The controlled version of U_π can then be constructed as

$$\mathrm{CU}_\pi := I \otimes |0\rangle\langle 0| + R_Y(\pi) \otimes |1\rangle\langle 1|, \tag{95}$$

where q_0 is the control qubit and q_1 is the target qubit. Equivalently, in the vector representation, the controlled-unitary looks as

$$\mathrm{CU}_\pi = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{96}$$

Consequently, the system quantum state at the second barrier of the circuit shown in Fig. 9 becomes

$$|\Psi_{\text{after}}\rangle = \mathrm{CU}_\pi \; |\Psi_{\text{before}}\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ i \\ -i \\ 1 \end{pmatrix}. \tag{97}$$

In order to rationalize the previous result, let us consider that, when the control is active, as in the second state (corresponding to $|01\rangle$) and in the fourth state (corresponding to $|11\rangle$), then the unitary U_π is applied to the target, leading to

$$U_\pi \; |y-\rangle = R_Y(\pi)\,|y-\rangle = e^{i\pi/2}\,|y-\rangle = i\,|y-\rangle, \tag{98}$$

which is equivalent to multiply by i the corresponding state before the controlled-unitary (i.e. $1\,i = i$ for the second state and $-i\,i = -i^2 = 1$ for the fourth state). Surprisingly, the system quantum state given by the vector reported in Eq. (97) is separable

$$|\Psi_{\text{after}}\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ i \\ -i \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ -i \end{pmatrix} \otimes \begin{pmatrix} 1 \\ i \end{pmatrix} = |y-\rangle \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}, \tag{99}$$

where the multiplying factor i moves to the control qubit, which is somehow counterintuitive and is the main point here. When a controlled-unitary acts on an eigenstate of the unitary gate, the eigenvalue's phase factor appears as a relative phase on the control qubit, leaving the target unchanged. In other words, one can say that the phase accumulates on the control qubit. This effect, called "kickback" effect, can be made even more evident by recalling that

$$\begin{aligned} |\Psi_{\text{after}}\rangle &= \frac{1}{2}\left(|00\rangle + i\,|01\rangle - i\,|10\rangle + |11\rangle\right) \\ &= \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle}{\sqrt{2}} + i\,\frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \otimes \frac{|1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} \\ &= |y-\rangle \otimes \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}. \end{aligned} \tag{100}$$

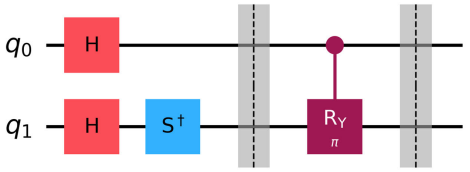

FIGURE 9. Example for explaining the phase kickback phenomenon. For clarity, horizontal lines represent quantum wires which correspond to qubits in the circuit, orange squares are the Hadamard gates H, blue square is the Hermitian adjoint of the phase gate $S^\dagger$, red square is the R_Y gate (see Eq. (86)) and red dot represents the control point in the controlled gate.

TABLE 1. Numerical tests for error analysis.

N	N_c	E_λ^{UB} [%]	$E_\lambda^{\max}$ [%]	$\mathbb{M}_e$ [%]
8	64	2.38	0.56	0.06
16	256	1.76	1.18	0.71
32	1024	1.61	1.16	0.30

In the table, $N = 2^n$, where n denotes the number of solution qubits, whereas $N_c = 2^{n_c}$, where n_c denotes the number of clock qubits. Throughout the analysis, we assume $N_c = N^2$, which is equivalent to $n_c = 2\,n$. The analytical upper bound for the error is given by Eq. (106), while the maximum error over all eigenvalues is defined by Eq. (102). Finally, the mean relative nodal error is evaluated according to Eq. (107).

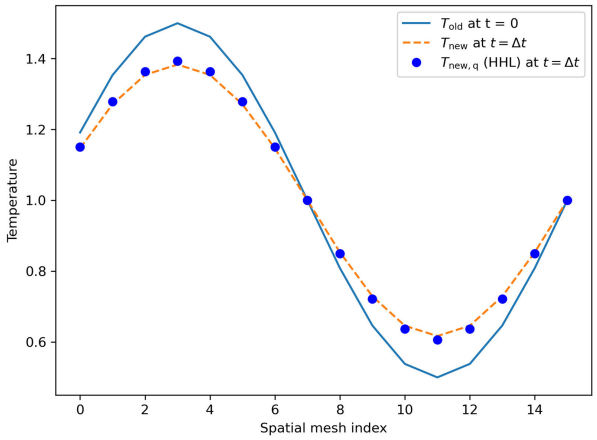

FIGURE 10. One time-step update of the temperature profile according to heat conduction equation by quantum computing by HHL algorithm: $n = 4$ input qubits, $n_c = 8$ clock qubits and 1 ancilla qubit. The meaning of the lines and dots is reported in the caption of Fig. 5.

Again, in the previous formula, the kickback effect modifies the control state but leaves the target state unchanged. This phase kickback mechanism constitutes the core operation of the Quantum Phase Estimation (QPE) stage of the HHL algorithm, where eigenvalue information is transferred to the clock register and subsequently extracted through the inverse quantum Fourier transform.

C. ERROR ANALYSIS

Error analysis aims at deriving *a priori* error estimates that can be used to guide the selection of the number of qubits allocated to the solution and clock registers, respectively. In the HHL algorithm, the accuracy of the computed solution is ultimately determined by the accuracy of the eigenvalue

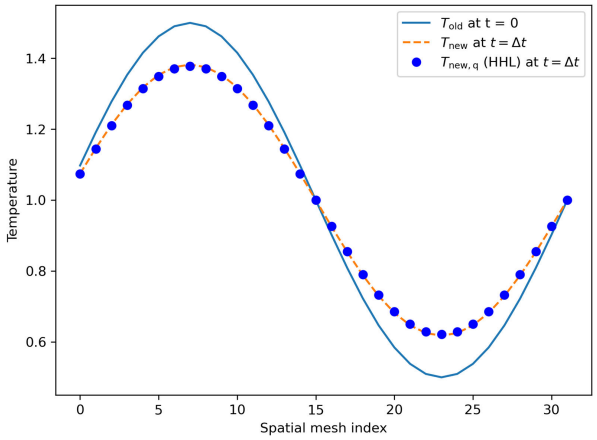

FIGURE 11. One time-step update of the temperature profile according to heat conduction equation by quantum computing by HHL algorithm: $n = 5$ input qubits, $n_c = 10$ clock qubits and 1 ancilla qubit. The meaning of the lines and dots is reported in the caption of Fig. 5.

estimation step. For the sake of simplicity, we first introduce the relative error $E_{\lambda,\,\mathrm{j}}$ associated with the estimation of the j-th eigenvalue λ_j^{Γ}. This error quantifies the discrepancy between the exact eigenvalue λ_j^{Γ} and its approximation λ_j^{SIM}, namely

$$E_{\lambda,\,\mathrm{j}} = \frac{\left|\lambda_j^{\Gamma} - \lambda_j^{\mathrm{SIM}}\right|}{\lambda_j^{\Gamma}}. \tag{101}$$

An overall estimate of the eigenvalue approximation error is provided by

$$E_{\lambda}^{\max} = \max_{j\in\{0,\,1\}^n} E_{\lambda,\,\mathrm{j}}. \tag{102}$$

Let us now derive an upper bound for the previous error. By recalling Eq. (82), it can be readily shown that the following inequality holds for all eigenvalues:

$$\left|\lambda_j^{\Gamma} - \lambda_j^{\mathrm{SIM}}\right| \leq \frac{1}{2\,(N_c - 1)}. \tag{103}$$

Since the relative error is largest for the smallest eigenvalue, namely λ_0^{Γ} given by Eq. (18), the previous estimate yields the following upper bound:

$$E_{\lambda}^{\max} \leq E_{\lambda}^{\mathrm{UB}} = \frac{1}{2\,(N_c - 1)}\,\frac{1}{\lambda_0^{\Gamma}} = \frac{1}{2}\left(\frac{4\,r + 1}{N_c - 1}\right). \tag{104}$$

Recalling that $r = D\,\Delta t/\Delta z^2$ is the (dimensionless) Fourier number and that $\Delta z = L/N$, it follows that

$$r = \frac{D\,\Delta t}{L^2}\,N^2 = r_0\,N^2, \tag{105}$$

where $r_0 = D\,\Delta t/L^2$ is a problem-dependent constant. In the present case, we assume $r_0 = 1/128$. By combining the previous results, the following analytical upper bound is obtained:

$$E_{\lambda}^{\mathrm{UB}} = \frac{1}{2}\left(\frac{4\,r_0\,N^2 + 1}{N_c - 1}\right). \tag{106}$$

In the following error analysis, we assume $N_c = N^2$. This choice is motivated by the fact that it yields a finite asymptotic upper bound for the eigenvalue estimation error, namely $\lim_{N\to\infty} E_{\lambda}^{\mathrm{UB}} = 2\,r_0$.

Regardless of the numerical methodology employed, the ultimate objective is to minimize the mean relative nodal error, defined as

$$\mathbb{M}_e = \frac{1}{N}\sum_j \frac{\left|T_j^{+} - \overline{T}_j^{+}\right|}{\overline{T}_j^{+}}, \tag{107}$$

where $\overline{T}_j^{+}$ denotes the reference nodal value (ground truth) [7]. The mean relative nodal error is reported in Table 1 as a function of $N = 2^n = \sqrt{N_c}$. It should be noted that the first row corresponds to $N = 8$ and $N_c = 64 = 2^6$, which is larger than the clock register considered in the main text ($N_c = 2^5$). This choice was made to satisfy the scaling relation $N_c = N^2$ adopted in the present error analysis. First, the proposed upper bound is confirmed to be valid, since the inequality $E_{\lambda}^{\max} \leq E_{\lambda}^{\mathrm{UB}}$ is satisfied in all cases. Although the upper bound exhibits an overall decreasing trend with increasing N, the behavior is not strictly monotonic. For example, in the case $N = 8$, the binarized eigenvalue approximation happens to provide an especially accurate representation of the exact eigenvalues, resulting in a smaller error than might otherwise be expected. This non-monotonic behavior is also reflected in the mean relative nodal error $\mathbb{M}_e$ of the computed solution. Nevertheless, the solution errors remain significantly smaller than the corresponding eigenvalue estimation errors, indicating that the HHL solution is relatively robust with respect to inaccuracies in the eigenvalue approximation. The resulting numerical solutions exhibit excellent agreement with the reference solution, as illustrated in Fig. 10 for $N = 16$ and Fig. 11 for $N = 32$.

REFERENCES

[1] M. Brooks, "Quantum computers: What are they good for?" *Nature*, vol. 617, no. 7962, pp. S1–S3, May 2023. [Online]. Available: https://www.nature.com/articles/d41586-023-01692-9

[2] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nat. Comput. Sci.*, vol. 2, no. 9, pp. 567–576, Sep. 2022. [Online]. Available: https://www.nature.com/articles/s43588-022-00311-3

[3] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nat. Rev. Phys.*, vol. 3, no. 9, pp. 625–644, Sep. 2021. [Online]. Available: https://www.nature.com/articles/s42254-021-00348-9

[4] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational quantum linear solver," *Quantum*, vol. 7, p. 1188, Nov. 2023. [Online]. Available: https://quantum-journal.org/papers/q-2023-11-22-1188/

[5] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014. [Online]. Available: https://www.nature.com/articles/ncomms5213

[6] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, "The variational quantum eigensolver: A review of methods and best practices," *Phys. Rep.*, vol. 986, pp. 1–128, Nov. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157322003118

[7] P. Asinari, M. M. Piredda, G. Barletta, P. De Angelis, N. Alghamdi, G. Trezza, M. Provenzano, M. Fasano, and E. Chiavazzo, "Leveraging quantum computing for heat conduction analysis: A case study in thermal engineering," *Case Stud. Thermal Eng.*, vol. 79, Mar. 2026, Art. no. 107813. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214157x26001759

[8] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, Oct. 2009, Art. no. 150502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502

[9] M. E. S. Morales, L. Pira, P. Schleich, K. Koor, P. C. S. Costa, D. An, A. Aspuru-Guzik, L. Lin, P. Rebentrost, and D. W. Berry, "Quantum linear system solvers: A survey of algorithms and applications," 2024, *arXiv:2411.02522*.

[10] J.-W. Pan, "Quantum technologies need big investments to deliver on their big promises," *Nature*, vol. 638, no. 8052, p. 862, Feb. 2025. [Online]. Available: https://www.nature.com/articles/d41586-025-00564-8

[11] 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. Modern Phys.*, vol. 94, no. 1, pp. 015004-1–015004-69, 2022. [Online]. Available: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.94.015004

[12] Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on IBM quantum experience," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Mar. 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-41324-9

[13] R. Yalovetzky, P. Minssen, D. Herman, and M. Pistoia, "Solving linear systems on quantum hardware with hybrid HHL++," *Sci. Rep.*, vol. 14, no. 1, pp. 1–16, Sep. 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-69077-0

[14] A. Zaman, H. J. Morrell, and H. Y. Wong, "A step-by-step HHL algorithm walkthrough to enhance understanding of critical quantum computing concepts," *IEEE Access*, vol. 11, pp. 77117–77131, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10189828

[15] L. Q. Galvão, A. B. M. de Souza, A. Oliveira S. Santos, A. Saimon S. Sousa, and C. Cruz, "Solving linear systems of equations with the quantum HHL algorithm: A tutorial on the physical and mathematical foundations for undergraduate students," 2025, *arXiv:2509.16640*.

[16] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with qiskit," 2024, *arXiv:2405.08810*.

[17] R. Zander, R. Seidel, and M. Petrič. (2025). *Solving Systems of Linear Equations via HHL Using Qrisp and Catalyst*. Accessed: Jun. 5, 2026. [Online]. Available: https://pennylane.ai/demos/linear_equations_hhl_qrisp_catalyst

[18] R. Seidel, S. Bock, N. V. Tcholtchev, and M. Hauswirth, "Qrisp: A framework for compilable high-level programming of gate-based quantum computers," in *Proc. 27th ACM SIGPLAN Int. Conf. Funct. Program.*, Ljubljana, Slovenia, 2022, pp. 11–16. [Online]. Available: https://publica.fraunhofer.de/handle/publica/445649

[19] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, and Y. Gu, "Quantum fast Poisson solver: The algorithm and complete and modular circuit design," *Quantum Inf. Process.*, vol. 19, no. 6, p. 170, Apr. 2020.

[20] (Feb. 2026). *QuantumThermal by Multi-Scale ModeLing Laboratory (SMaLL) at Politecnico Di Torino*. [Online]. Available: https://github.com/SMaLL-PoliTo/QuantumThermal

[21] M. Revythi and G. Koukiou, "Quantum machine learning and deep learning: Fundamentals, algorithms, techniques, and real-world applications," *Mach. Learn. Knowl. Extraction*, vol. 7, no. 3, p. 75, Aug. 2025.

[22] F. Turro, A. Lignarolo, and D. Dragoni, "Toward practical application of the quantum Carleman lattice Boltzmann method in industrial CFD simulations," *IEEE Trans. Quantum Eng.*, vol. 6, pp. 1–18, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11199306

[23] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig, "Quantum linear systems algorithms: A primer," 2018, *arXiv:1802.08227*.

[24] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, "A divide-and-conquer algorithm for quantum state preparation," *Sci. Rep.*, vol. 11, no. 1, p. 6329, Mar. 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-85474-1

[25] G. Agliardi and E. Prati, "Quantum data encoding as a distinct abstraction layer in the design of quantum circuits," *Quantum Sci. Technol.*, vol. 10, no. 2, Jan. 2025, Art. no. 025008, doi: 10.1088/2058-9565/ada6f8.

[26] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. U.K.: Cambridge University Press, Dec. 2010.

[27] T. G. Wong, *Introduction to Classical and Quantum Computing*. Omaha, NE, USA: Rooted Grove, 2022.

[28] D. Gottesman, "The Heisenberg representation of quantum computers," 1998, *arXiv:quant-ph/9807006*.

PIETRO ASINARI received the B.S. and M.S. degrees (cum laude) in mechanical engineering and the Ph.D. degree in energetics from Politecnico di Torino, Italy, in 2001 and 2005, respectively.

From 2020 to 2024, he was the Scientific Director with Italian National Institute of Metrological Research (INRiM). He is currently the President with INRiM and a Full Professor with the Department of Energy, Politecnico di Torino, where he also serves on the Board of the Ph.D. Program in Energetics. He has been the Co-Founder of the Multi-Scale Modelling Laboratory (SMaLL). His institutional activities focus on fostering cross-ministerial initiatives in strategic areas, particularly quantum technologies and energy security. He played a key role in the acquisition of the first IQM quantum computer in Italy, inaugurated in May 2025, and promoted INRiM's participation in the Master's Degree Program in Quantum Engineering with Politecnico di Torino, since 2023. In 2018, he was appointed as an Advisory Board Member of the OpenKIM Project, USA. He has been actively involved in European research initiatives, serving as a member for the Operational Management Board (OMB) of European Materials Modeling Council (EMMC), from 2014 to 2019. He has authored more than 140 peer-reviewed publications and holds some patents. His research interests include energy and environmental sustainability, renewable-driven water production, heat and mass transfer in energy systems, multiscale materials modeling, and advanced numerical methods, such as lattice Boltzmann and molecular dynamics simulations.

Prof. Asinari is a member of several professional societies, including Italian Union of Thermo-Fluid Dynamics (UIT), where he serves on the Executive Committee and as the Vice-President. His doctoral dissertation was awarded the ENI-Italgas Award for Energy and Environment.

MATTEO MARIA PIREDDA received the B.S. and M.S. degrees (cum laude) in mechanical engineering from Politecnico di Torino, Italy, where he is currently pursuing the Ph.D. degree in energetics (38th cycle) with the Department of Energy.

He has contributed to several peer-reviewed scientific publications in the fields of multiphase flow modeling, numerical simulation, and computational methods for energy systems. He also presented the work entitled "Quantum Computing for Solving Linear Systems of Equations in Engineering Applications: The Case Study of the Heat Conduction Equation" at an international scientific workshop. His research activities focus on computational fluid dynamics and advanced numerical methods for the simulation of complex physical systems. His work includes the development of lattice Boltzmann frameworks for multiphase flows based on Eulerian–Eulerian Navier–Stokes formulations, as well as investigations on heat transfer and transport phenomena in energy systems. His research also explores the application of emerging computational paradigms, including quantum computing, to problems in thermal and fluid engineering.

GIULIO BARLETTA received the B.S. degree in energy engineering and the M.S. degree (cum laude) in energy and nuclear engineering from Politecnico di Torino, Italy, in 2021 and 2023, respectively, and the M.S. degree in mechanical engineering from the University of Illinois Chicago, USA, in 2023. He is currently pursuing the Ph.D. degree in energetics with the Department of Energy, Politecnico di Torino.

He is conducting his doctoral research in collaboration with Italian National Research Council (CNR) and Polo Solare Organico della Regione Lazio (CHOSE). He has been involved in research on lithium-ion battery modeling, atmospheric water harvesting systems, and perovskite photovoltaics through collaborations with academic and research institutions in Italy and abroad. His research focuses on artificial intelligence, scientific machine learning, and advanced computational methods for energy applications. In particular, he works on data-driven approaches for the characterization, degradation analysis, and optimization of perovskite solar cells, including the use of spatially resolved luminescence imaging and multimodal learning frameworks. His research interests include surrogate modeling and the application of quantum computing methodologies to engineering and physics-based problems.

MATTEO FASANO received the Ph.D. degree in energy engineering and nanotechnology from the Scuola Interpolitecnica di Dottorato, in 2015.

He has held international research appointments as a Research Fellow with Houston Methodist Research Institute and a Visiting Researcher with Massachusetts Institute of Technology. He has collaborated with scientists from Imperial College London, the University of Minnesota, and Université Claude Bernard Lyon 1. He is currently an Associate Professor with Politecnico di Torino, Italy, where he is also the Co-Director of the Multi-Scale Modelling Laboratory (SMaLL) and a member of the Management Board of the Interdepartmental Clean Water Center Initiative. At Politecnico di Torino, he teaches courses in energy applications of materials, advanced engineering thermodynamics, and nanoscale heat and mass transfer modeling. Since 2014, he has co-authored more than 100 peer-reviewed publications, including articles in *Nature Communications*, *Nature Sustainability*, and *Science Advances*. He is involved in European projects on passive cooling materials, thermal recycling, and porous materials for energy storage and filtration, and collaborates with companies in the automotive, space, and energy sectors. His expertise includes atomistic, mesoscopic, continuum, and system-level modeling, as well as machine-learning methods. His research focuses on multiscale modeling of heat and mass transfer, with applications to enhanced heat transfer, thermal energy storage, solar energy harvesting, and thermal desalination.

Prof. Fasano's doctoral thesis was recognized as the Best Ph.D. Research by the Department of Energy, Politecnico di Torino. He received the ENI Award, in 2017.

ELIODORO CHIAVAZZO received the B.S. and M.S. degrees (summa cum laude) in mechanical engineering and the Ph.D. degree from ETH Zurich, with a thesis on reactive thermo-fluid dynamics.

He has held research positions as a fellow with Princeton University and a Visiting Scientist with the Technical University of Munich. He is currently a Full Professor of applied thermodynamics with Politecnico di Torino, where he has been also the Deputy Coordinator of mechanical engineering studies and a member of the Ph.D. Board in Energetics. He is the Director of the Multi-Scale Modeling Laboratory (SMaLL), leading a multidisciplinary research group working on advanced energy systems. He has authored over 120 peer-reviewed publications in leading international journals, including *Science Advances*, PNAS, and *Nature* family journals. He has delivered numerous invited talks at international conferences. He has coordinated and participated in several European and national research projects as a principal investigator and the work package leader, collaborating with major academic institutions and industry partners. He serves as a reviewer for major funding agencies and an editorial board member for international journals. His research focuses on multiscale modeling of heat and mass transfer phenomena, spanning atomistic to macroscopic approaches, with applications in energy materials and technologies. His work includes AI-driven materials discovery, thermal energy storage, electronic cooling, and solar-driven desalination and cooling systems.