

Step-by-Step HHL Algorithm Walkthrough to Enhance the Understanding of Critical Quantum Computing Concepts

Hector Jose Morrell Jr, Anika Zaman, and Hiu Yung Wong*

Department of Electrical Engineering,

San Jose State University, San Jose, CA 95192

(Dated: May 25, 2022)

arXiv:2108.09004v3 [quant-ph] 24 May 2022

Abstract

After learning basic quantum computing concepts, it is desirable to reinforce the learning using an important and relatively complex algorithm through which the students can observe and appreciate how the qubits evolve and interact with each other. Harrow-Hassidim-Lloyd (HHL) quantum algorithm, which can solve Linear System Problems with exponential speed-up over the classical method and is the basic of many important quantum computing algorithms, is used to serve this purpose. The HHL algorithm is explained analytically followed by a 4-qubit numerical example in bra-ket notation. Matlab code corresponding to the numerical example is available for students to gain a deeper understanding of the HHL algorithm from a pure matrix point of view. A quantum circuit programmed using qiskit is also provided which can be used for real hardware execution in IBM quantum computers. After going through the material, students are expected to have a better appreciation of the concepts such as basis transformation, bra-ket and matrix representations, superposition, entanglement, controlled operations, measurement, Quantum Fourier Transformation, Quantum Phase Estimation, and quantum programming. To help readers review these basic concepts, brief explanations augmented by the HHL numerical examples in the main text are provided in the Appendix.

I. INTRODUCTION

Quantum Computing is promising in solving challenging engineering [1], biomedical [2] and finance [3] problems. It has a tremendous advancement in the last two decades and, recently, quantum supremacy has been demonstrated using a 53-qubit system [4]. Therefore, the training of a quantum technology workforce is an imminent goal for many countries (e.g. [5]) to support this fast-growing industry.

However, quantum technology is based on concepts very different from our daily and classical experiences. In the early stage of learning quantum computing, although linking to daily and classical experience may enhance the understanding of certain quantum concepts and such an approach should not be de-emphasized, we believe a fast and robust way of training a quantum workforce is to train the students to be able to emulate a quantum processor and trace the evolution of the qubits. This is particularly useful in learning

* hiuyung.wong@sjsu.edu

quantum algorithms without a quantum mechanics background. Such an approach obviates the students from cognitive conflicts, which can be resolved later, if possible, after they understand how quantum computing works. This also embraces the “Shut up and calculate!” approach proposed by Mermin on how to deal with the uncomfortable feeling toward quantum mechanics interpretation [6].

Besides analytical equations, matrix representation and computer simulations are important tools to enhance the understanding of qubit evolution. However, available examples that include computer simulations are usually of simple algorithms and, very often, without matrix representation. There is a lack of examples of important and relatively complex algorithms which combine some of the most important quantum computing concepts and basic algorithms. Such examples are desirable to allow students to appreciate the roles and the interplay of various basic concepts in a more realistic quantum algorithm. Harrow-Hassidim-Lloyd (HHL) quantum algorithm [7][8] which can be used to solve linear system problems (LSP) and can provide exponential speedup over the classical conjugate gradient method [9] is chosen for this purpose. HHL is the basic of many more advanced algorithms and is important in various applications such as machine learning [10] and modeling of quantum systems [2][11]. It has also been proposed to solve the Poisson equation [12][13]. In this paper, we detail the qubit evolution in Harrow-Hassidim-Lloyd (HHL) quantum algorithm analytically with a 4-qubit circuit as a numerical example. Although HHL examples are available elsewhere (e.g. [14][15]), this paper has certain characteristics which are not all found in those examples. Firstly, the HHL algorithm is discussed analytically step-by-step and is self-contained. Secondly, a numerical example is given in bra-ket notation mirroring the analytical equations. Thirdly, a Matlab code corresponding exactly to the numerical example is available to enhance the understanding from a matrix point of view. The Matlab code allows the students to trace how the wavefunction evolves instead of just seeing the magnitudes of the coefficients as in IBM-Q. Fourthly, a qiskit code written in python [16] corresponding to the numerical example is available and can be run in IBM simulation and hardware machines [17]. Finally, in the example, all the 4 qubits are traced throughout the process without simplification.

The readers are assumed to have the following background concepts which are further enhanced through the step-by-step walkthrough of the HHL algorithm: basis transformation, bra-ket and matrix representations, superposition, entanglement, encoding, controlled op-

erations, measurement, Quantum Fourier Transformation, and quantum programming. To make this paper self-contained and to help the readers better appreciate the roles of these basic concepts in the HHL, an Appendix is devoted to briefly explaining these concepts using the examples from the main text. A more detailed explanation of these concepts using a similar approach as in this paper can be found in [18].

A. How to use this paper

For readers who have a fresh memory of the basic concepts, they can start reading from Section II, in which the HHL algorithm is discussed step-by-step analytically followed by a numerical example in Section III. The basic concepts mentioned in the Appendix are referred to in the main text and readers are encouraged to review them when needed.

For readers who need reviews on the basic concepts first, they are encouraged to go over the Appendix first before reading the main text.

For readers who have devoted substantial time to learning HHL elsewhere but just need a numerical example to reinforce the understanding, they might start with the numerical example in Section III.

Equations in the Appendix begin with 'V'. If the equations are examples from the main text, the same equation number is used in the Appendix.

II. HHL ALGORITHM

A. Definitions and Overview

We will first give an overview of the problem and the HHL algorithm. Details will be discussed in the following subsections with reference to the Appendix for reviewing basic concepts.

A linear system problem (LSP) can be represented as the following

$$A\vec{x} = \vec{b} \tag{1}$$

where A is an $N_b \times N_b$ Hermitian matrix and \vec{x} and \vec{b} are N_b -dimensional vectors. For simplicity, it is assumed $N_b = 2^{n_b}$. Otherwise, dummy equations can be added to convert

the system satisfy this assumption. A and \vec{b} are known and \vec{x} is the unknown to be solved, i.e.

$$\vec{x} = A^{-1}\vec{b} \quad (2)$$

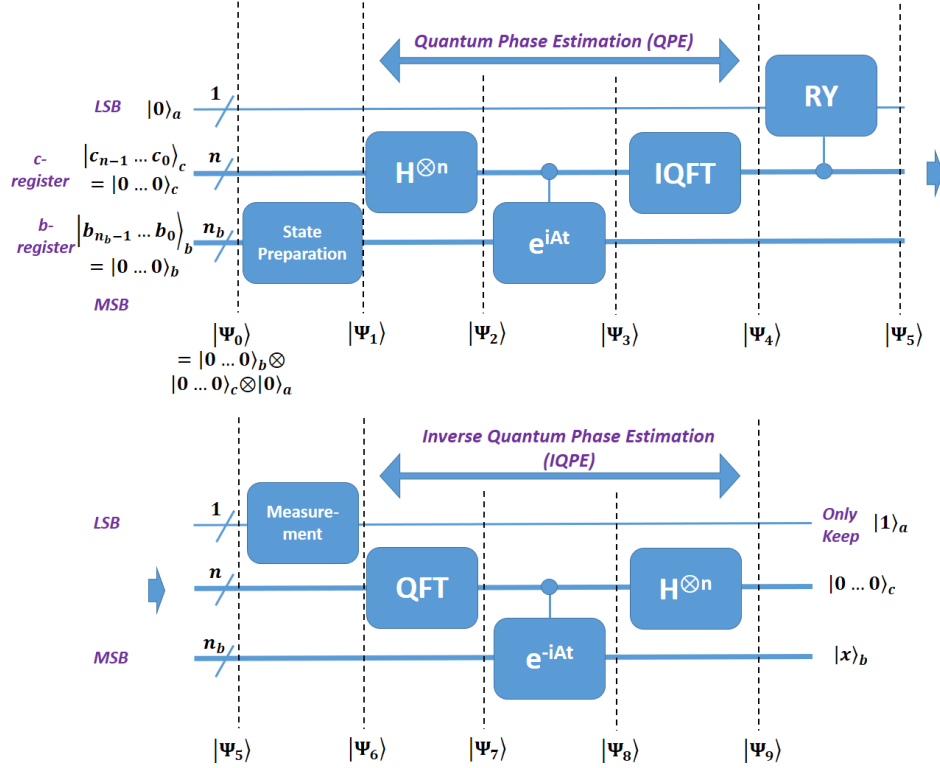


FIG. 1. Schematic of the HHL quantum circuit flowing from left to right. The circuit is decomposed into the top and bottom portions for clarity. Note that the lowest qubit in the diagram is the most significant bit (MSB) while the top one is the least significant bit (LSB).

As an example, $A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}$, $\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and $\vec{x} = \begin{pmatrix} \frac{3}{8} \\ \frac{9}{8} \end{pmatrix}$ with $n_b = 1$ and $N_b = 2^1 = 2$. Readers may refer to Appendix V 13 and Appendix V 14 to review how a LSP is solved classically using Gaussian Elimination and Conjugate Gradient Method, respectively.

A is assumed to be Hermitian (See Appendix V 1). If it is not Hermitian, the A can be converted to $\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$, which is Hermitian. Readers may refer to [19] for the more advanced treatment when A is not Hermitian.

Figure 1 shows the schematic of the HHL algorithm and the corresponding circuit to

solve an LSP. In the HHL quantum algorithm, the N_b components of \vec{b} and \vec{x} are encoded as the amplitudes/coefficients (*amplitude encoding*) of basis states of the n_b -qubits, $|\rangle_b$, which form a \mathbb{C}^{N_b} Hilbert space. These n_b qubits are called b-register. n_b is chosen to be large enough to encode \vec{b} , i.e. 2^{n_b} needs to be the same as the length of the vectors \vec{b} and \vec{x} . The matrix A is simulated through *Hamiltonian encoding* by encoding it as the Hamiltonian of a unitary gate. Appendix V 9 reviews examples of various encoding schemes.

The HHL algorithm has 5 main components, namely state preparation, quantum phase estimation (QPE), ancilla bit rotation, inverse quantum phase estimation (IQPE), and measurement. In this paper, the little-endian convention is used. In a little-endian convention, the rightmost (ending) qubit represents the least significant bit (LSB). For example, in a 4-qubit system, $|0001\rangle$ in binary is $|1\rangle$ in decimal because the 1 in the basis state $|0001\rangle$ is the LSB, representing 2^0 instead of 2^3 (if it were the most significant bit, MSB). Moreover, in the circuit diagrams, the lowest qubit represents the MSB and the topmost qubit represents the LSB, which is a convention used in qiskit [16] and the IBM-Q platform [17].

As shown in Figure 1, besides the b-register, which belongs to the more significant bits, there are two more sets of inputs to the algorithm. The first set is sometimes called the c-register because it is related to the time (clock) in the controlled rotation in the QPE part. Therefore, they are also called the clock qubits. C-register stores the values of the phase of the eigenvalues of the A matrix after the QPE. There are n qubits in the c-register. Since *basis encoding* is used (i.e. the phase value is encoded as the basis number (See Appendix V 9), the value of n determines how accurate the phase can be stored. A larger n results in higher accuracy when the encoding is not exact. We set $N = 2^n$.

The last set of qubits is the ancilla qubit $|\rangle_a$ which is the LSB. The ancilla qubit, as its name implies, is important to help achieve the goal although it will be discarded at the end, as will be detailed later.

The matrix A , which is a Hamiltonian, may be written as a linear combination of the outer products of its eigenvectors, $|u_i\rangle\langle u_i|$ weighted by its eigenvalues, λ_i (See Appendix V 8).

$$A = \sum_{i=0}^{2^{n_b}-1} \lambda_i |u_i\rangle\langle u_i| \quad (3)$$

Since A is diagonal in its eigenvector basis, its inverse is simply, $A^{-1} = \sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} |u_i\rangle\langle u_i|$.

\vec{b} can be also expressed in the basis formed by the eigenvectors of A, such that

$$|b\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle \quad (4)$$

Therefore, Eq. (2) can be encoded as,

$$|x\rangle = A^{-1} |b\rangle = \sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} b_i |u_i\rangle \quad (5)$$

by using the fact that $\langle u_i | u_j \rangle = \delta_{ij}$. The goal of the HHL algorithm is to find the solution in this form and $|x\rangle$ is stored in the b-register. Eq. (5) shows that if the right hand side, $\sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} b_i |u_i\rangle$, is stored in the b-register, it is equivalent to storing $|x\rangle$ in the b-register. But the solution is encoded as the amplitudes of the basis vectors $|0\rangle / |1\rangle$ (the measurement basis). Therefore, the solutions are not $\lambda_i^{-1} b_i$, which are the amplitudes of the eigenvector basis vectors. However, one will naturally get the correct amplitudes if it is measured in the $|0\rangle / |1\rangle$ basis and this is only possible if the qubits are not entangled with other qubits. This can be checked mathematically replacing $|u_i\rangle$ by $|0\rangle / |1\rangle$ based on their relationship. If $|u_i\rangle$ are entangled with other qubits, one cannot obtain the desired solution. This will be clear after the ancilla bit rotation to be detailed later.

The equation needs also to be prepared so that the eigenvectors, $|u_i\rangle$, and $|b\rangle$ are normalized so they can be properly represented as a unit vector in quantum computing. Therefore, Eq. (4) and Eq. (5) require

$$\sum_{j=0}^{2^{n_b}-1} |b_j|^2 = 1 \quad (6)$$

$$\sum_{i=0}^{2^{n_b}-1} |\lambda_i^{-1} b_i|^2 = 1 \quad (7)$$

B. State Preparation

There are total $n_b + n + 1$ qubits and they are initialized as

$$|\Psi_0\rangle = |0 \cdots 0\rangle_b |0 \cdots 0\rangle_c |0\rangle_a = |0\rangle^{\otimes n_b} |0\rangle^{\otimes n} |0\rangle \quad (8)$$

In the state preparation, $|0 \cdots 0\rangle_b$ in the b-register needs to be rotated to have the amplitudes correspond to the coefficients of \vec{b} . That is

$$\vec{b} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N_b-1} \end{pmatrix} \Leftrightarrow \beta_0 |0\rangle + \beta_1 |1\rangle + \cdots + \beta_{N_b-1} |N_b - 1\rangle = |b\rangle \quad (9)$$

The vector \vec{b} is represented in a column form on the left with coefficients β' s. This is also a valid representation of $|b\rangle$. On the right, the corresponding basis of the Hilbert space formed by the n_b qubits is written explicitly. Therefore,

$$|\Psi_1\rangle = |b\rangle_b |0 \cdots 0\rangle_c |0\rangle_a \quad (10)$$

From now on, some of the subscripts of the kets will be omitted when there is no ambiguity. Since the state preparation depends on the actual value of \vec{b} , it will be discussed in more detail in the numerical example.

C. Quantum Phase Estimation

Quantum phase estimation (QPE) is also an eigenvalue estimation algorithm. QPE has three components, namely the superposition of the clock qubits through Hadamard gates, controlled rotation, and Inverse Quantum Fourier Transform (IQFT). The goal of QPE is to estimate the phase of the eigenvalues of the unitary rotation matrix, $U = e^{iAt}$, in the controlled gate, $C - U$, (Fig. 1) used in the QPE. Again, this gate encodes the matrix A as its Hamiltonian. It is also instructive to note that the eigenvalues of U must be unity (i.e. in the form of $e^{i\theta}$) as U is unitary. Therefore, the phase of the eigenvalue of the gate is proportional to the eigenvalue of A . As a result, by using QPE in the HHL algorithm, it is expected the eigenvalues of A will be encoded in the c-register after the QPE, i.e. at $|\Psi_4\rangle$. As it will be clear later, the eigenvalues are only encoded through basis encoding. The c-register does not store the exact eigenvalues.

Here we assume the readers are already familiar with IQFT and it will not be explained in detail. Readers may review the basic concepts in Appendix V 11 and V 12.

In the first step of QPE, Hadamard gates are applied to the clock qubits to create a

superposition of the clock qubits,

$$|\Psi_2\rangle = I^{\otimes n_b} \otimes H^{\otimes n} \otimes I |\Psi_1\rangle \quad (11)$$

$$= |b\rangle \frac{1}{2^{\frac{n}{2}}} (|0\rangle + |1\rangle)^{\otimes n} |0\rangle \quad (12)$$

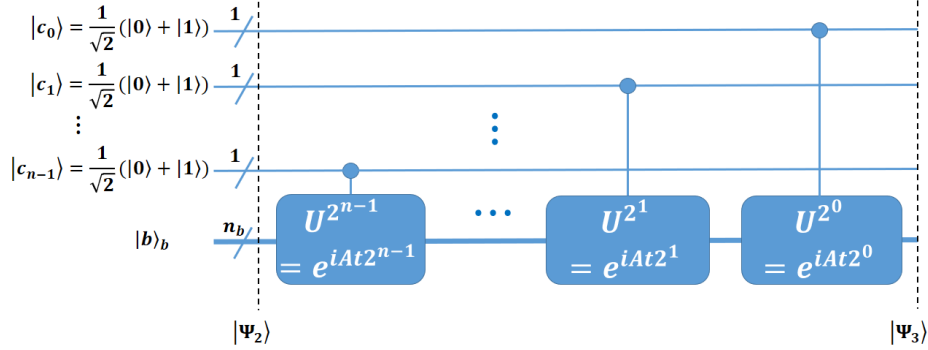


FIG. 2. The controlled-rotation part of QPE. U is replaced by e^{iAt} in the HHL algorithm.

In the controlled rotation part, controlled gates are applied to $|b\rangle$ with the clock qubits as the controlling qubits (Figure 2). The number of the qubit, n , of the c-register determines the number of the controlled gates. The gates are in the form of U^{2^r} , where r is the index of the clock qubit. Also, $U = e^{iAt}$. For the most significant bit in the c-register, $|c_{n-1}\rangle$, it controls the gate $U^{2^{n-1}}$ on the b-register while the least significant one, $|c_0\rangle$, controls the gate $U^{2^0} = U$ on the b-register.

To understand how it works, we begin by assuming that $|b\rangle$ is an eigenvector of U with eigenvalue $e^{2\pi i\phi}$. The eigenvalue is written in this form so that the phase, ϕ , will be encoded as the basis state in the c-register (See Eq. (16)). Therefore, based on the definition of eigenvalues and eigenvectors (See Section V 8),

$$U |b\rangle = e^{2\pi i\phi} |b\rangle \quad (13)$$

When the control clock qubit is $|0\rangle$, $|b\rangle$ will not be affected. If the clock bit is $|1\rangle$, U will be applied to $|b\rangle$. This is equivalent to multiplying $e^{2\pi i\phi 2^j}$ in front of the $|1\rangle$ of the j -th clock qubit, $|c_j\rangle$, as one can assign the prefactor to the controlling qubit. Therefore, after the controlled- U operation, we have

$$\begin{aligned}
|\Psi_3\rangle &= |b\rangle \otimes \left(\frac{1}{2^{\frac{n}{2}}} (|0\rangle + e^{2\pi i \phi 2^{n-1}} |1\rangle) \otimes (|0\rangle + e^{2\pi i \phi 2^{n-2}} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i \phi 2^0} |1\rangle) \right) \otimes |0\rangle_a \\
&= |b\rangle \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} |k\rangle |0\rangle_a
\end{aligned} \tag{14}$$

In the IQFT part, only the clock qubits are affected. Note that in certain literature, this is called Quantum Fourier Transform (QFT) (Appendix V 11).

$$\begin{aligned}
|\Psi_4\rangle &= |b\rangle IQFT \left(\frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} |k\rangle \right) |0\rangle_a \\
&= |b\rangle \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} (IQFT |k\rangle) |0\rangle_a \\
&= |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} \left(\sum_{y=0}^{2^n-1} e^{-2\pi i y k / N} |y\rangle \right) |0\rangle_a \\
&= \frac{1}{2^n} |b\rangle \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i k (\phi - y/N)} |y\rangle |0\rangle_a
\end{aligned} \tag{15}$$

Due to *interference*, only for $|y\rangle$ satisfying the condition $\phi - y/N = 0$ will have a finite amplitude of $\sum_{k=0}^{2^n-1} e^0 = 2^n$. Otherwise, the amplitude is $\sum_{k=0}^{2^n-1} e^{2\pi i k (\phi - y/N)} = 0$ due to *destructive interference*. By ignoring the states with zero amplitude, we may rewrite $|\Psi_4\rangle$ as

$$\begin{aligned}
|\Psi_4\rangle &= \frac{1}{2^n} |b\rangle \sum_{k=0}^{2^n-1} e^{2\pi i k \cdot 0} |N\phi\rangle |0\rangle_a \\
&= |b\rangle |N\phi\rangle |0\rangle_a
\end{aligned} \tag{16}$$

Therefore, in QPE, the clock qubits are used to represent the phase information of U , which is ϕ , and the accuracy depends on the number of qubits, n .

Since in *Hamiltonian encoding*, U is related to A through

$$U = e^{iAt} \tag{17}$$

where t is the evolution time for that Hamiltonian. U is obviously also diagonal in A 's eigenvector, $|u_i\rangle$, basis. If $|b\rangle = |u_j\rangle$,

$$U |b\rangle = e^{i\lambda_j t} |u_j\rangle \tag{18}$$

By equating $i\lambda_j t$ to $2\pi i\phi$ in Eq. (13), we get $\phi = \lambda_j t/2\pi$ and Eq. (16) becomes

$$|\Psi_4\rangle = |u_j\rangle |N\lambda_j t/2\pi\rangle |0\rangle_a \quad (19)$$

Thus the eigenvalues of A have been encoded in the clock qubits (*basis encoding*). However, in general, $|b\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle$ (Eq. 4), by using *superposition*,

$$|\Psi_4\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |N\lambda_j t/2\pi\rangle |0\rangle_a \quad (20)$$

λ_j are usually not integers. We will choose t so that $\tilde{\lambda}_j = N\lambda_j t/2\pi$ are integers. Therefore, the encoded values $\tilde{\lambda}_j$ is usually different from λ_j but its scaled version.

Ψ_4 can be rewritten as

$$|\Psi_4\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle |0\rangle_a \quad (21)$$

D. Controlled Rotation and Measurement of the Ancilla Qubit

The next step is to rotate the ancilla qubit, $|0\rangle_a$, based on the encoded eigenvalues in the c-register, such that

$$|\Psi_5\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle_a + \frac{C}{\tilde{\lambda}_j} |1\rangle_a \right) \quad (22)$$

where C is a constant. This is the goal and we will show why this is useful.

When the ancilla qubit is measured, the ancilla qubit *wavefunction* will *collapse* to either $|0\rangle$ or $|1\rangle$. If it is $|0\rangle$, the result will be discarded and the computation will be repeated until the measurement is $|1\rangle$. Therefore, the final wavefunction of interest is

$$|\Psi_6\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\tilde{\lambda}_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle \frac{C}{\tilde{\lambda}_j} |1\rangle_a \quad (23)$$

where the prefactor is due to normalization after measurement. Since $|\frac{C}{\tilde{\lambda}_j}|^2$ is the probability of obtaining $|1\rangle$ when the ancilla bit is measured, C should be chosen to be as large as possible. Compared to Eq. (5), the result resembles the answer $|x\rangle$ that we are looking for. However, we can only obtain the correct result if the b-register is measured in the

eigenvector basis (i.e. $|u_j\rangle$ instead of $|0\rangle / |1\rangle$). However, the b-register is *entangled* with the clock qubits, $|\tilde{\lambda}_j\rangle$. This means that we cannot factorize the result into a tensor product of the c-register and b-register (See the discussion after Eq. (5) and Appendix V 6). As a result, we cannot convert the b-register into the $|0\rangle / |1\rangle$ measurement basis with the desired amplitudes. We will need to uncompute the state so that it gives the right results in the $|0\rangle / |1\rangle$ measurement during which the b-register and c-register will be *unentangled*.

The measurement of the ancilla qubit can be and is usually performed after uncomputation. However, since the ancilla bit is not involved in any operations after the controlled rotation, measuring the ancilla bit before the uncomputation gives the same result. For simplicity in the derivation, it is thus performed before the uncomputation.

E. Uncomputation - Inverse QPE

The uncomputation is done by using Inverse QPE. Firstly, QFT is applied to the clock qubits.

$$\begin{aligned} |\Psi_7\rangle &= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} |\frac{b_j C}{\tilde{\lambda}_j}|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\tilde{\lambda}_j} |u_j\rangle QFT |\tilde{\lambda}_j\rangle |1\rangle_a \\ &= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} |\frac{b_j C}{\tilde{\lambda}_j}|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\tilde{\lambda}_j} |u_j\rangle \left(\frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i y \tilde{\lambda}_j / N} |y\rangle \right) |1\rangle_a \end{aligned} \quad (24)$$

Then inverse controlled-rotations of the b-register by the clock qubits are applied with $U^{-1} = e^{-iAt}$. Similar to the forward process, when the controlling $r - th$ clock qubit is $|0\rangle$, $|u_j\rangle$ will not be affected. If the $r - th$ clock qubit is $|1\rangle$, $(U^{-1})^{2^r}$ will be applied to $|u_j\rangle$. This is equivalent to multiplying $e^{-i\lambda_j t y}$ if the c-register is $|y\rangle$. This is due to the similar argument in Eq. (14) and the fact that $2\pi i \phi = i\lambda_j t$. Therefore,

$$|\Psi_8\rangle = \frac{1}{2^{n/2} \sqrt{\sum_{j=0}^{2^{n_b}-1} |\frac{b_j C}{\tilde{\lambda}_j}|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\tilde{\lambda}_j} |u_j\rangle \left(\sum_{y=0}^{2^n-1} e^{-i\lambda_j t y} e^{2\pi i y \tilde{\lambda}_j / N} |y\rangle \right) |1\rangle_a \quad (25)$$

Since we have set $\tilde{\lambda}_j = N\lambda_j t / 2\pi$, therefore, the two exponential terms cancel each other and

$$\begin{aligned}
|\Psi_8\rangle &= \frac{1}{2^{n/2} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a \\
&= \frac{1}{2^{n/2} \frac{Nt}{2\pi} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a \\
&= \frac{C}{2^{n/2} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} |x\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a
\end{aligned} \tag{26}$$

The clock qubits and the b-register are now *unentangled* and the b-register stores $|x\rangle$. By applying the Hadamard gate on the clock qubits, finally, we have

$$\begin{aligned}
|\Psi_9\rangle &= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle |0\rangle_c^{\otimes n} |1\rangle_a \\
&= \frac{C}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} |x\rangle_b |0\rangle_c^{\otimes n} |1\rangle_a
\end{aligned} \tag{27}$$

If C is real and by using Eq. (7),

$$|\Psi_9\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j}{\lambda_j} \right|^2}} |x\rangle_b |0\rangle_c^{\otimes n} |1\rangle_a \tag{28}$$

$$= |x\rangle_b |0\rangle_c^{\otimes n} |1\rangle_a \tag{29}$$

Here, the solution $|x\rangle$ (Eq. 5) is stored in the b-register successfully.

III. NUMERICAL EXAMPLE

We will present a numerical example and apply HHL to it step-by-step. The implementation is shown in Figure 3. Firstly, we will discuss how to implement the controlled- U and ancilla qubit rotations.

A. Encoding Scheme

In this example, the matrix A and vector \vec{b} are set to be

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \quad (30)$$

$$\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (31)$$

The eigenvectors of A are $\vec{u}_0 = \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$, $\vec{u}_1 = \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ with eigenvalues $e_0 = \frac{2}{3}$ and $e_1 = \frac{4}{3}$ respectively. We need to use *basis encoding* to encode the eigenvalues in the basis formed by the clock qubit and 2 qubits are needed by encoding e_0 as $|01\rangle$ and e_1 as $|10\rangle$ so that it maintains the ratio of $e_1/e_0 = 2$. This means $\tilde{\lambda}_0 = 1$ and $\tilde{\lambda}_1 = 2$ or in other words, $|\tilde{\lambda}_0\rangle = |01\rangle$ and $|\tilde{\lambda}_1\rangle = |10\rangle$. This gives a perfect encoding with $n = 2$ (i.e. $N = 4$). Therefore, t is chosen to be $\frac{3\pi}{4}$ to achieve the encoding scheme since $\tilde{\lambda}_j = N\lambda_j t/2\pi$.

Since \vec{b} is a 2-dimensional complex vector, it can be encoded using 1 qubit and, thus, $n_b = 1$.

The solution to the LSP is found to be

$$\vec{x} = \begin{pmatrix} \frac{3}{8} \\ \frac{9}{8} \end{pmatrix} \quad (32)$$

whereby, the ratio of $|x_0|^2$ to $|x_1|^2$ is 1 : 9.

B. $C-U$ Implementation

In reality, we expect the controlled- U operation to be implemented by a physical system with a Hamiltonian the same as A . However, to understand the algorithm, we will derive the matrix for U and then map this to the CU gate used in IBM-Q. Since $n = 2$, there are two operations needed, namely $U^{2^1} = U^2$ and $U^{2^0} = U$, controlled by c_1 and c_0 , respectively.

In order to find the corresponding matrix for $U^2 = e^{i2At}$ and $U = e^{iAt}$, we need to perform *similarity transformation* on $i2At$ and iAt , exponentiate then, and transform back to the original basis.

The transformation matrix from the original basis to the eigenvector basis is

$$V = \begin{pmatrix} \vec{u}_0 & \vec{u}_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \quad (33)$$

Since V is real, its Hermitian conjugate, V^\dagger equals itself.

The diagonalized A , i.e. expressed in the basis formed by \vec{u}_0 and \vec{u}_1 , is

$$A_{diag} = V^\dagger A V = \begin{pmatrix} \frac{2}{3} & 0 \\ 0 & \frac{4}{3} \end{pmatrix} \quad (34)$$

As it is diagonal, U can be obtained by exponentiating the elements accordingly.

$$\begin{aligned} U_{diag} &= \begin{pmatrix} e^{ie_0 t} & 0 \\ 0 & e^{ie_1 t} \end{pmatrix} \\ &= \begin{pmatrix} e^{i\pi/2} & 0 \\ 0 & e^{i\pi} \end{pmatrix} \\ &= \begin{pmatrix} i & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \quad (35)$$

where $t = 3\pi/4$ as mentioned earlier is used. Also,

$$U_{diag}^2 = U_{diag} U_{diag} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad (36)$$

It is worth noting that both are naturally *unitary* which is a requirement for a quantum operation.

To obtain U and U^2 in the original basis, we need to apply similarity transformation again in the reverse direction,

$$\begin{aligned}
U &= VU_{diag}V^\dagger \\
&= \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} i & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} -1+i & 1+i \\ 1+i & -1+i \end{pmatrix}
\end{aligned} \tag{37}$$

$$\begin{aligned}
U^2 &= VU_{diag}^2V^\dagger \\
&= \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \\
&= \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}
\end{aligned} \tag{38}$$

To implement U and U^2 , a 4-parameter arbitrary unitary gate with global phase [18],

$$U = \begin{pmatrix} e^{i\gamma} \cos(\theta/2) & -e^{i(\gamma+\lambda)} \sin(\theta/2) \\ e^{i(\gamma+\phi)} \sin(\theta/2) & e^{i(\gamma+\phi+\lambda)} \cos(\theta/2) \end{pmatrix} \tag{39}$$

By choosing $\theta = \pi, \phi = \pi, \lambda = 0, \gamma = 0$, U^2 is implemented.

By choosing $\theta = \pi/2, \phi = -\pi/2, \lambda = \pi/2, \gamma = 3\pi/4$, U is implemented.

For the IQPE part, we also need to implement U^{-1} and U^{-2} . Since in this example, $(U^2)^{-1} = U^2$, one can use the same set of parameters to implement $(U^2)^{-1}$.

However,

$$U^{-1} = \frac{1}{2} \begin{pmatrix} -1-i & 1-i \\ 1-i & -1-i \end{pmatrix} \tag{40}$$

We need to choose $\theta = \pi/2, \phi = \pi/2, \lambda = -\pi/2, \gamma = -3\pi/4$ to implement U^{-1} .

The controlled version of matrix U' can then be constructed using

$$C - U' = I \otimes |0\rangle\langle 0| + U' \otimes |1\rangle\langle 1| \tag{41}$$

Note that in this equation, only the controlling clock bit and the b-register are included for simplicity. For example,

$$\begin{aligned}
C - U^{-1} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1-i & 1-i \\ 1-i & -1-i \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1-i & 0 & 1-i \\ 0 & 0 & 0 & 0 \\ 0 & 1-i & 0 & -1-i \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1-i & 0 & 1-i \\ 0 & 0 & 2 & 0 \\ 0 & 1-i & 0 & -1-i \end{pmatrix}
\end{aligned} \tag{42}$$

C. Implementataion of the Controlled-Rotation of Ancilla Qubit

The coefficients of $|0\rangle$ and $|1\rangle$ of the ancilla bit after rotation in Eq. (22) are $\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}}$ and $\frac{C}{\tilde{\lambda}_j}$, respectively. The sum of the square of the magnitudes of the coefficients is 1 as required. This means also $C \leq \tilde{\lambda}_j$. Since the minimal $\tilde{\lambda}_j$ is 1, we will set $C = 1$ to maximize the probability of measuring $|1\rangle$ during the ancilla bit measurement.

The transformation of $|0\rangle_a$ to $\sqrt{1 - \frac{1}{\tilde{\lambda}_j^2}} |0\rangle_a + \frac{1}{\tilde{\lambda}_j} |1\rangle_a$ is known to be equivalent to $RY(\theta)$ rotation,

$$RY(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \tag{43}$$

with $\theta = 2 \arcsin(\frac{1}{\tilde{\lambda}_j})$. One can check this by multiplying $RY(\theta)$ to $|0\rangle_a$. Therefore, we will establish a function to implement this rotation and this function only need to be valid when the input are the encoded eigenvalues because only encoded eigenvalues have zero magnitudes in the c-register as shown in Eq. (21). The function is defined as

$$\theta(c) = \theta(c_1 c_0) = 2 \arcsin(\frac{1}{c}) \tag{44}$$

where c is the value of the clock qubits and c_1c_0 is its binary form.

Since only $|\tilde{\lambda}_j\rangle$ has non-zero amplitude in Eq. (21), we only need to set up Eq. (44) such that it is correct for $|c\rangle=|01\rangle$ and $|10\rangle$, namely

$$\theta(1) = \theta(01) = 2 \arcsin\left(\frac{1}{1}\right) = \pi \quad (45)$$

$$\theta(2) = \theta(10) = 2 \arcsin\left(\frac{1}{2}\right) = \frac{\pi}{3} \quad (46)$$

The following function can achieve the goal,

$$\theta(c) = \theta(c_1c_0) = \frac{\pi}{3}c_1 + \pi c_0 \quad (47)$$

Therefore, the controlled rotation can be implemented as

$$|1\rangle\langle 1| \otimes I \otimes RY\left(\frac{\pi}{3}\right) + |0\rangle\langle 0| \otimes I \otimes I + I \otimes |1\rangle\langle 1| \otimes RY(\pi) + I \otimes |0\rangle\langle 0| \otimes I \quad (48)$$

where the operators operate on qubits $|c_1\rangle$, $|c_0\rangle$, and $|a\rangle$ from left to right, respectively.

D. Quantum Circuit

An HHL circuit for the numerical example is then built and shown in Figure 3. We will then walk through the circuit using numerical substitution.

E. Numerical Substitution

The algorithm begins with

$$|\Psi_0\rangle = |0\rangle_b \otimes |00\rangle_c \otimes |0\rangle_a = |0000\rangle \quad (49)$$

X-gate is then applied to convert $|0\rangle_b$ to $|1\rangle_b$ with

$$|\Psi_1\rangle = X \otimes I \otimes I |\Psi_0\rangle = |1000\rangle \quad (50)$$

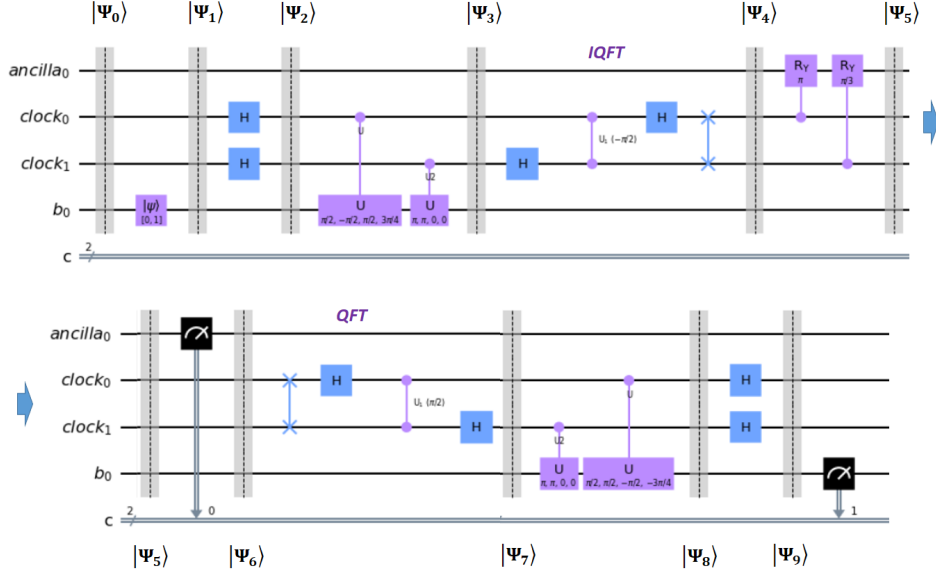


FIG. 3. The HHL circuit corresponding to the numerical example built to run in IBM-Q. This circuit is partitioned in the same way as the general HHL schematic in Figure 1.

After applying the Hadamard gates to create a superposition among the clock qubits,

$$\begin{aligned}
 |\Psi_2\rangle &= I \otimes H^{\otimes n} \otimes I |\Psi_1\rangle \\
 &= |1\rangle \frac{1}{2^{\frac{n}{2}}} (|0\rangle + |1\rangle)^{\otimes 2} |0\rangle \\
 &= \frac{1}{2} (|1000\rangle + |1010\rangle + |1100\rangle + |1110\rangle)
 \end{aligned} \tag{51}$$

Before applying the CU3 gates in the bra-ket notation, it will be convenient to **perform a basis change to the eigenvector basis of A** . Since $|1\rangle = \frac{1}{\sqrt{2}}(-|u_0\rangle + |u_1\rangle)$, we have $b_0 = \frac{-1}{\sqrt{2}}$ and $b_1 = \frac{1}{\sqrt{2}}$. Therefore,

$$\begin{aligned}
 |\Psi_2\rangle &= |1\rangle \frac{1}{2} (|000\rangle + |010\rangle + |100\rangle + |110\rangle) \\
 &= \frac{1}{\sqrt{2}} (-|u_0\rangle + |u_1\rangle) \frac{1}{2} (|000\rangle + |010\rangle + |100\rangle + |110\rangle) \\
 &= \frac{1}{2\sqrt{2}} (-|u_0000\rangle - |u_0010\rangle - |u_0100\rangle - |u_0110\rangle \\
 &\quad + |u_1000\rangle + |u_1010\rangle + |u_1100\rangle + |u_1110\rangle)
 \end{aligned} \tag{52}$$

In the controlled rotation operations, when the corresponding c-register is $|k\rangle_c$, a phase change of $\phi_j = k\lambda_j t/2\pi$ is added (i.e. multiplied by $e^{2\pi i \phi_j}$) for $|u_j\rangle$. Since $t = \frac{3\pi}{4}$, $\lambda_0 = \frac{2}{3}$

and $\lambda_1 = \frac{4}{3}$, we have

$$\begin{aligned}
|\Psi_3\rangle &= \frac{1}{2\sqrt{2}}(-|u_0000\rangle - e^{2\pi i\phi_0}|u_0010\rangle - e^{2\pi i2\phi_0}|u_0100\rangle - e^{2\pi i3\phi_0}|u_0110\rangle \\
&\quad + |u_1000\rangle + e^{2\pi i\phi_1}|u_1010\rangle + e^{2\pi i2\phi_1}|u_1100\rangle + e^{2\pi i3\phi_1}|u_1110\rangle) \\
&= \frac{1}{2\sqrt{2}}(-|u_0000\rangle - e^{i\lambda_0 t}|u_0010\rangle - e^{i2\lambda_0 t}|u_0100\rangle - e^{i3\lambda_0 t}|u_0110\rangle \\
&\quad + |u_1000\rangle + e^{i\lambda_1 t}|u_1010\rangle + e^{i2\lambda_1 t}|u_1100\rangle + e^{i3\lambda_1 t}|u_1110\rangle) \\
&= \frac{1}{2\sqrt{2}}(-|u_0000\rangle - e^{i\pi/2}|u_0010\rangle - e^{i\pi}|u_0100\rangle - e^{i3\pi/2}|u_0110\rangle \\
&\quad + |u_1000\rangle + e^{i\pi}|u_1010\rangle + e^{i2\pi}|u_1100\rangle + e^{i3\pi}|u_1110\rangle) \\
&= \frac{1}{2\sqrt{2}}(-|u_0000\rangle - i|u_0010\rangle + |u_0100\rangle + i|u_0110\rangle \\
&\quad + |u_1000\rangle - |u_1010\rangle + |u_1100\rangle - |u_1110\rangle)
\end{aligned} \tag{53}$$

Before applying IQFT, the terms are regrouped for simplicity.

$$\begin{aligned}
|\Psi_3\rangle &= \frac{1}{2\sqrt{2}}((-|u_0\rangle + |u_1\rangle)|00\rangle + (-i|u_0\rangle - |u_1\rangle)|01\rangle \\
&\quad + (|u_0\rangle + |u_1\rangle)|10\rangle + (i|u_0\rangle - |u_1\rangle)|11\rangle)|0\rangle
\end{aligned} \tag{54}$$

Now apply IQFT to the clock qubits, e.g.

$$\begin{aligned}
IQFT|10\rangle &= IQFT|2\rangle \\
&= \frac{1}{2^{2/2}} \sum_{y=0}^{2^2-1} e^{-2\pi i 2y/4} |y\rangle \\
&= \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle) \\
&= \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)
\end{aligned} \tag{55}$$

Similarly,

$$IQFT|00\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \tag{56}$$

$$IQFT|01\rangle = \frac{1}{2}(|00\rangle - i|01\rangle - |10\rangle + i|11\rangle) \tag{57}$$

$$IQFT|11\rangle = \frac{1}{2}(|00\rangle + i|01\rangle - |10\rangle - i|11\rangle) \tag{58}$$

Therefore, applying IQFT to $|\Psi_3\rangle$ and substituting Eq. (55) to (58),

$$\begin{aligned}
|\Psi_4\rangle &= IQFT |\Psi_3\rangle \\
&= \frac{1}{4\sqrt{2}} \\
&\quad ((-|u_0\rangle + |u_1\rangle)(|00\rangle + |01\rangle + |10\rangle + |11\rangle) + \\
&\quad (-i|u_0\rangle - |u_1\rangle)(|00\rangle - i|01\rangle - |10\rangle + i|11\rangle) + \\
&\quad (|u_0\rangle + |u_1\rangle)(|00\rangle - |01\rangle + |10\rangle - |11\rangle) + \\
&\quad (i|u_0\rangle - |u_1\rangle)(|00\rangle + i|01\rangle - |10\rangle - i|11\rangle)) |0\rangle \\
&= \frac{1}{\sqrt{2}}(-|u_0\rangle |01\rangle + |u_1\rangle |10\rangle) |0\rangle
\end{aligned} \tag{59}$$

It can be seen that after IQFT, the eigenvalues are encoded in the clock qubits as $|01\rangle$ and $|11\rangle$ with non-zero amplitudes due *constructive interference*. $b_0 = \frac{-1}{\sqrt{2}}$ and $b_1 = \frac{1}{\sqrt{2}}$. We clearly see the entanglement between the b-register and the c-register that $|u_0\rangle$ goes with $|01\rangle$ and $|u_1\rangle$ goes with $|11\rangle$.

After performing the ancilla qubit rotation,

$$\begin{aligned}
|\Psi_5\rangle &= \sum_{j=0}^{2^1-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right) \\
&= -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle \left(\sqrt{1 - \frac{1}{1^2}} |0\rangle + \frac{1}{1} |1\rangle \right) + \\
&\quad \frac{1}{\sqrt{2}} |u_1\rangle |10\rangle \left(\sqrt{1 - \frac{1}{2^2}} |0\rangle + \frac{1}{2} |1\rangle \right)
\end{aligned} \tag{60}$$

If the measurement of the ancilla bit is $|1\rangle$,

$$|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle |10\rangle |1\rangle \right) \tag{61}$$

Applying QFT to the encoded eigenvalues, we have

$$\begin{aligned}
QFT |10\rangle &= QFT |2\rangle \\
&= \frac{1}{2^{2/2}} \sum_{y=0}^{2^2-1} e^{2\pi i 2y/4} |y\rangle \\
&= \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle)
\end{aligned} \tag{62}$$

$$\begin{aligned}
QFT |01\rangle &= QFT |1\rangle \\
&= \frac{1}{2} (|00\rangle + i|01\rangle - |10\rangle - i|11\rangle)
\end{aligned} \tag{63}$$

Therefore, applying QFT to $|\Psi_6\rangle$ and substituting Eq. (62) to (63), we obtain

$$|\Psi_7\rangle = \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle \frac{1}{2} (|00\rangle + i|01\rangle - |10\rangle - i|11\rangle) |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) |1\rangle \right) \quad (64)$$

For the controlled rotation, the state is multiplied by $e^{-i\lambda_j t}$ and $e^{-i2\lambda_j t}$ if $c_0 = 1$ and $c_1 = 1$, respectively. Since $e^{-i\lambda_0 t} = -i$, $e^{-i2\lambda_0 t} = -1$, $e^{-i\lambda_1 t} = -1$, $e^{-i2\lambda_1 t} = 1$, and $Nt/2\pi = 3/2$

$$\begin{aligned} |\Psi_8\rangle &= \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) |1\rangle \right) \\ &= \frac{1}{2} \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle \right) (|00\rangle + |01\rangle + |10\rangle + |11\rangle) |1\rangle \\ &= \frac{1}{2} \left(\frac{2}{3} \right) \sqrt{\frac{8}{5}} \left(-\frac{1}{\frac{2}{3}\sqrt{2}} |u_0\rangle + \frac{1}{\frac{4}{3}\sqrt{2}} |u_1\rangle \right) (|00\rangle + |01\rangle + |10\rangle + |11\rangle) |1\rangle \end{aligned} \quad (65)$$

Finally, by applying Hardamard gate to the clock qubits,

$$|\Psi_9\rangle = \frac{2}{3} \sqrt{\frac{8}{5}} \left(-\frac{1}{\frac{2}{3}\sqrt{2}} |u_0\rangle + \frac{1}{\frac{4}{3}\sqrt{2}} |u_1\rangle \right) |00\rangle |1\rangle \quad (66)$$

It can be verified that $|\Psi_9\rangle$ is a normalized vector as it should be because every operation in the HHL circuit is unitary and preserves the norm.

Eq. (66) can be simplified by substituting $|u_0\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle$ and $|u_1\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$. We obtain,

$$|\Psi_9\rangle = \frac{1}{2} \sqrt{\frac{2}{5}} (|0\rangle + 3|1\rangle) |00\rangle |1\rangle \quad (67)$$

The probability ratio of obtaining $|0\rangle$ and $|1\rangle$ when b-register is measured is thus 1 : 9 as expected.

F. Simulation Results

Matlab code implementing the numerical example using matrix approach is created and available at [20]. In the Matlab code, measurement is not performed (i.e. not partial tracing

of the matrix). Ψ_9 is found to be

$$\begin{pmatrix} -0.4330 \\ 0.2500 \\ 0.0000 \\ -0.0000 \\ 0.0000 \\ -0.0000 \\ 0.0000 \\ 0.0000 \\ 0.4330 \\ 0.7500 \\ -0.0000 \\ 0.0000 \\ -0.0000 \\ 0.0000 \\ -0.0000 \\ 0.0000 \end{pmatrix} \quad (68)$$

Since $|0\rangle_c$ are discarded during the measurement step, only $|0001\rangle$ and $|1001\rangle$ are left. Their amplitude ratio is $0.25^2 : 0.75^2 = 1 : 9$ as expected.

The circuit in Fig. 3 is also simulated in the IBM-Q system (code available at [20]). Since only the b-register and the ancilla qubit are measured, there are only four possible outputs as shown in Figure 4. Again, only $|1\rangle_a$ should be considered. The ratio of the measurement probability of $|0\rangle_b |1\rangle_a$ to $|1\rangle_b |1\rangle_a$ is $0.063 : 0.564 = 1 : 8.95$, which is close to the expected value.

On the other hand, due to the imperfection and noise in a real quantum computer, the hardware execution of the same circuit does not give a satisfactory result (Figure 5). The ratio of the measurement probability of $|0\rangle_b |1\rangle_a$ to $|1\rangle_b |1\rangle_a$ is only $0.142^2 : 0.361^2 = 1 : 2.54$.

IV. CONCLUSION

In this paper, we presented the HHL algorithm through a step-by-step walkthrough of the derivation. A numerical example is also presented in the bra-ket notation. The numerical

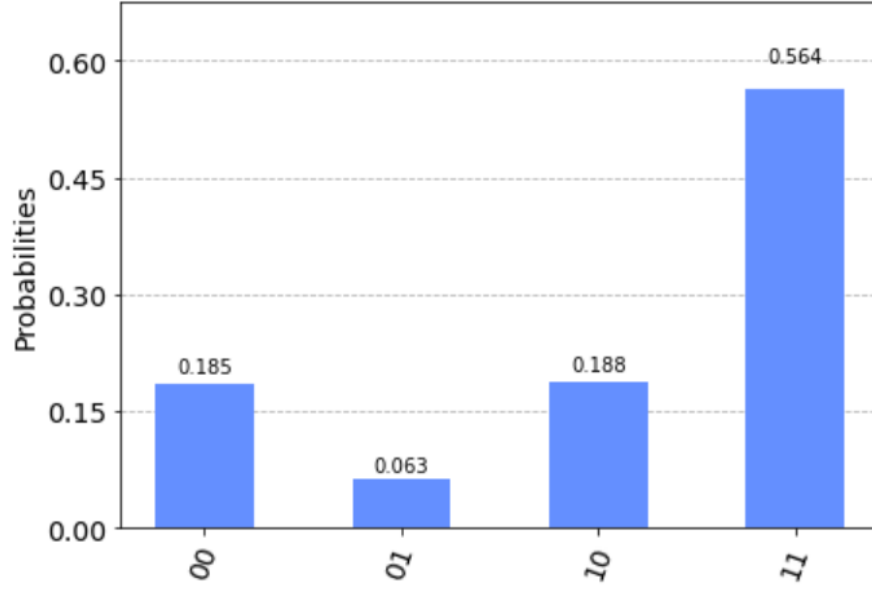


FIG. 4. Simulation result of the circuit in Figure 3 using IBM-Q. Only the MSB, $|\rangle_b$ and the LSB, $|\rangle_a$ are measured.

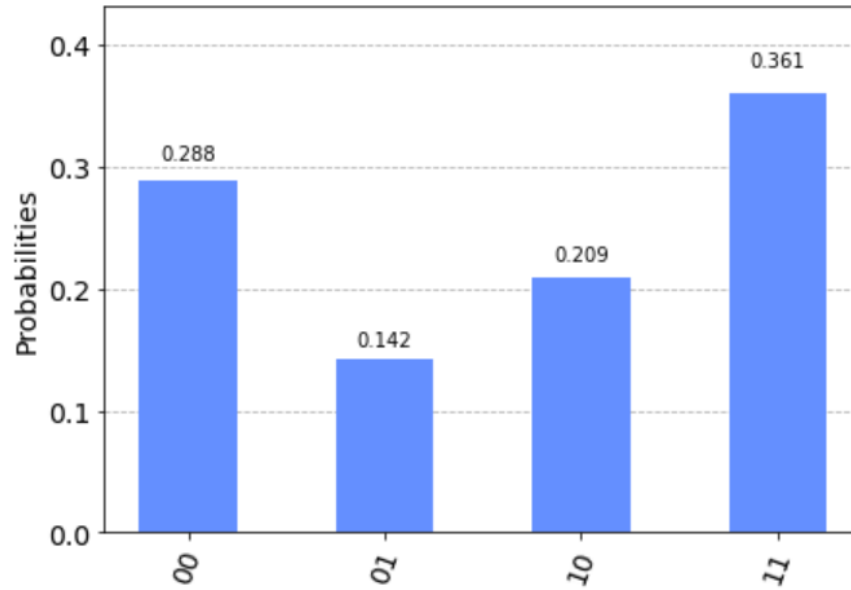


FIG. 5. Hardware result of the circuit in Figure 3 run in machine ibmq_santiago. Only the MSB, $|\rangle_b$ and the LSB, $|\rangle_a$ are measured.

example echos the analytical derivation to help students understand how qubits evolve in

this important and relatively complex algorithm. A Matlab code corresponding to the numerical example is constructed to help understand the algorithm from the matrix point of view. Qiskit circuit of the corresponding circuit which can be simulated in IBM-Q and run on their quantum computing hardware is also available. Through this self-contained and step-by-step walkthrough, the basic concepts in quantum computing are reinforced.

V. APPENDIX

1. Hermitian matrix

A Hermitian matrix is a matrix that equals to its adjoint matrix (transpose followed by complex conjugation). That is, if A is a Hermitian matrix, then it is defined as,

$$A = A^\dagger = (A^T)^* \quad (\text{V.69})$$

where A^T is the transpose of A .

In this paper, the matrix, A , in the LPS to be solved is assumed to be Hermitian.

Another example is in Eq. (33), where V is Hermitian.

$$\begin{aligned} V &= \begin{pmatrix} \vec{u}_0 & \vec{u}_1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \end{aligned} \quad (33)$$

2. Bra-ket Notation

Bra-ket notation is commonly used in Quantum Mechanics. A vector \vec{v} is represented as $|v\rangle$ in its ket form. The bra form of the vectors forms a dual space to the space of the kets. The bra form of \vec{v} is $\langle v|$.

In matrix representation, ket is the complex conjugate transpose of bra and vice versa. For example, if $|v\rangle = \begin{pmatrix} 1 \\ -i \end{pmatrix}$, then $\langle v| = (1 \ i.)$

3. *Superposition*

Superposition or Quantum Superposition is a quantum state which is the linear combination of two or more basis states. For example, a superposition state can be $|v\rangle = c_1 |1\rangle + c_2 |0\rangle$, where c_1 and c_2 are complex number and $|1\rangle$ and $|0\rangle$ are basis states. Hadamard gate is the gate commonly used to create a superposition state (Appendix V 5).

4. *Basis Transformation and Quantum Gate*

In quantum computing, we only care about the basis transformation due to rotation in the hyperspace. The transformation is equivalent to the multiplication of the basis vectors by a unitary matrix, U , which is the transformation matrix. All quantum gates can be defined as how the basis vectors are transformed from the initial basis vector to the final basis vectors. Usually, a quantum gate rotates a basis state into another basis state (e.g. the NOT gate) and has its classical counterpart. But there are some gates that rotate a basis state to a superposition of two or more basis states. Such gates have no classical counterparts. For example, a Hadamard gate defines how an initial basis vector is rotated to an equal superposition of two basis vectors (Appendix V 5).

5. *Hadamard Gate*

The Hadamard gate is a quantum gate that does not have a classical counterpart. It rotates the basis state to create an equal superposition of the basis states. For a 1-qubit case, this means it has equal probability (i.e. $\frac{1}{2}$) of measuring $|0\rangle$ and $|1\rangle$.

The matrix form of the Hadamard gate is,

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{V.70}$$

When it is applied on the basis state $|1\rangle$, which is $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ in matrix form, we have,

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (\text{V.71})$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (\text{V.72})$$

which can also be represented in bra-ket form as,

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (\text{V.73})$$

In this paper, Hadamard gates are applied in clock qubit to create superposition from $|\psi_1\rangle$ to $|\psi_2\rangle$. For example, in Eq (11),

$$|\Psi_2\rangle = I^{\otimes n_b} \otimes H^{\otimes n} \otimes I |\Psi_1\rangle \quad (11)$$

where $|\psi_2\rangle$ is obtained by applying tensor product of Identity gates and an n -qubit Hadamard gate to $|\psi_1\rangle$. The Identity gates are applied to the b-register and the ancilla qubit while the Hadamard gate is applied to the clock qubits. In this equation, the n -qubit Hadamard gate is represented as $H^{\otimes n}$, i.e. the tensor product of n 1-qubit Hadamard gates.

6. Entanglement

Entanglement refers to the quantum state of a 2- or more-qubit system that cannot be expressed as a tensor product of the individual qubit. This is an important feature that quantum computing uses often. As an example,

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (\text{V.74})$$

is an entangled state. It cannot be expressed as a tensor product of two individual qubit states.

In this paper, after the ancilla bit rotation, we have

$$|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle |10\rangle |1\rangle \right) \quad (61)$$

where the b-register and the c-register are entangled. $|u_0\rangle$ ($|u_1\rangle$) always appears with $|01\rangle$ ($|10\rangle$) after the measurement.

If the b-register *were* not entangled with the c-register, we have

$$|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} |u_0\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle \right) \quad (\text{V.75})$$

By substituting $|u_0\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle$ and $|u_1\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$ and after simplification, we have

$$\begin{aligned} |\Psi_6\rangle &= \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} \left(\frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle \right) + \frac{1}{2\sqrt{2}} \left(\frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \right) \\ &= \sqrt{\frac{8}{5}} \left(\frac{1}{4} |0\rangle + \frac{3}{4} |1\rangle \right) \end{aligned} \quad (\text{V.76})$$

This is the same as Eq. (67). The probability of measuring $|0\rangle$ and $|1\rangle$ has the ratio of 1:9 as expected.

However, when there is entanglement, the probability of measuring $|0\rangle$ and $|1\rangle$ would not be 1:9 because the previous grouping is impossible.

$$|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left(-\frac{1}{\sqrt{2}} \left(\frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle \right) |01\rangle + \frac{1}{2\sqrt{2}} \left(\frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) |10\rangle \right) \quad (\text{V.77})$$

7. Controlled Operation

Controlled operation requires more than one qubit. For a 2-qubit controlled gate, an operation is applied to a qubit (the target qubit), if the value of the controlling qubit is 1 in the basis vector.

For example, in Figure 2, $|b\rangle$ is the target qubit and $|c_{n-1}\rangle$ is the controlling qubit. The operation of $U^{2^{n-1}}$ is applied to $|b\rangle$ only if $|c_{n-1}\rangle$ is 1 in the basis state (e.g. $|bc_{n-1}\dots\rangle = |01\dots\rangle$).

In general, the controlled version of a unitary gate, U' , can be implemented using the following equation if the LSB is the controlling qubit.

$$C - U' = I \otimes |0\rangle\langle 0| + U' \otimes |1\rangle\langle 1| \quad (41)$$

which literally means that if the controlling qubit is 0, Identity gate is applied to the target qubit (MSB). Otherise, U' is applied to the target qubit.

8. Eigenvalue and Eigenvector

When a non-zero $n \times n$ matrix A is applied to an n -dimensional vector \vec{V} and has the following relationship,

$$A\vec{V} = \lambda\vec{V} \quad (\text{V.78})$$

where λ is a scalar, then, by definition, \vec{V} and λ are the eigenvector and eigenvalue of A , respectively. In Eq. (3), the matrix A is expressed as a linear combination of the outer products of its eigenvectors, $|u_i\rangle \langle u_i|$.

$$A = \sum_{i=0}^{2^{n_b}-1} \lambda_i |u_i\rangle \langle u_i| \quad (3)$$

This can be checked by applying A to its eigenvector $|u_j\rangle$,

$$\begin{aligned} A|u_j\rangle &= \sum_{i=0}^{2^{n_b}-1} \lambda_i |u_i\rangle \langle u_i| |u_j\rangle \\ &= \sum_{i=0}^{2^{n_b}-1} \lambda_i |u_i\rangle \delta_{ij} \\ &= \lambda_j |u_j\rangle \end{aligned} \quad (\text{V.79})$$

which meets the definition in Eq. (V.78).

9. Different Types of Encoding

The three common types of encodings are explained here.

- **Basis Encoding-** Basis encoding converts classical information such as numbers or matrix to quantum information in the form of basis states. For example,

$$x = 2 \xrightarrow{\text{binary}} 10 \xrightarrow{\text{quantum-state}} |10\rangle \quad (\text{V.80})$$

$$x = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \xrightarrow{\text{binary}} \begin{pmatrix} 10 \\ 11 \end{pmatrix} \xrightarrow{\text{quantum-state}} |1011\rangle \quad (\text{V.81})$$

- **Amplitude Encoding-** Amplitude encoding encodes the information as the coefficients of the basis vectors. For example, for

$$\vec{v} = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \quad (\text{V.82})$$

which is assumed to be normalized ($|v| = 1$), it can be encoded as in the following quantum state,

$$|v\rangle = v_0 |0\rangle + v_1 |1\rangle \quad (\text{V.83})$$

where v_0 and v_1 become the coefficients of the basis states, $|0\rangle$ and $|1\rangle$, respectively. In the main text, Eq. (9) shows how the values of the components of vector $|b\rangle$ are encoded using amplitude encoding.

- **Hamiltonian Encoding-** One type of the Hamiltonian encoding is to encode the matrix as the Hamiltonian in a unitary gate. For example, in this paper, Eq. (17) shows that

$$U = e^{iAt} \quad (17)$$

where it encodes matrix A as the Hamiltonian of the unitary gate U . A needs to be Hermitian as it is used to represent the Hamiltonian (the energy) of the system.

10. Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT) transforms an N -dimensional vector \vec{X} to another N -dimensional vector \vec{Y} . The transformation matrix Ω contains the powers of the N -th root of unity, $\omega = e^{i2\pi/N}$. The transformation is represented as,

$$\vec{Y} = \Omega \vec{X}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega^{-0 \cdot 0} & \omega^{-0 \cdot 1} & \dots & \omega^{-0 \cdot (N-1)} \\ \omega^{-1 \cdot 0} & \omega^{-1 \cdot 1} & \dots & \omega^{-1 \cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{-(N-1) \cdot 0} & \omega^{-(N-1) \cdot 1} & \dots & \omega^{-(N-1) \cdot (N-1)} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \quad (\text{V.84})$$

11. Inverse Quantum Fourier Transform (IQFT) and Quantum Fourier Transform(QFT)

Mathematically, IQFT is similar to DFT (See Appendix V 10). The transformation matrix, U_I , is $N \times N$ for an N -dimensional Hilbert space. Therefore, $N = 2^n$ for an n -qubit system. Eq. (V.84) becomes

$$|Y\rangle = U_I |X\rangle \quad (\text{V.85})$$

And U_I has the same expression as Ω in DFT.

$$U_I = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega^{-0 \cdot 0} & \omega^{-0 \cdot 1} & \dots & \omega^{-0 \cdot (N-1)} \\ \omega^{-1 \cdot 0} & \omega^{-1 \cdot 1} & \dots & \omega^{-1 \cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{-(N-1) \cdot 0} & \omega^{-(N-1) \cdot 1} & \dots & \omega^{-(N-1) \cdot (N-1)} \end{pmatrix} \quad (\text{V.86})$$

Note that in some literature, e.g. [18], this form of IQFT is called QFT. $|X\rangle$ and $|Y\rangle$ are the quantum states/vectors in the N -dimensional Hilbert space. IQFT can be treated as the rotation of $|X\rangle$ to $|Y\rangle$.

If $|X\rangle$ is a basis vector $|k\rangle$, applying IQFT to $|k\rangle$ using Eq. (V.86), we have

$$U_I |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{-jk} |j\rangle \quad (\text{V.87})$$

This is the equation used often in this paper. It tells us that by applying IQFT to a basis vector, the basis vector is rotated to a superposition of all basis vectors weighted by the powers of the N -th root of unity.

For example, in Eq. (55) in the main text,

$$\begin{aligned} IQFT |10\rangle &= IQFT |2\rangle \\ &= \frac{1}{2^{2/2}} \sum_{y=0}^{2^2-1} e^{-2\pi i 2y/4} |y\rangle \\ &= \frac{1}{2} (|0\rangle - |1\rangle + |2\rangle - |3\rangle) \\ &= \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \end{aligned} \quad (\text{55})$$

where $N = 2^2 = 4$, $k = 2$, $j = y$ in Eq. (V.87) is used. The basis state $|10\rangle$ becomes a superposition of all other basis states after IQFT.

Another more complex example is the general equation, Eq.(15), for the IQFT in Figure 1.

$$\begin{aligned}
|\Psi_4\rangle &= |b\rangle \text{IQFT}\left(\frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} |k\rangle\right) |0\rangle_a \\
&= |b\rangle \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} (\text{IQFT} |k\rangle) |0\rangle_a \\
&= |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} \left(\sum_{y=0}^{2^n-1} e^{-2\pi i y k / N} |y\rangle \right) |0\rangle_a \\
&= \frac{1}{2^n} |b\rangle \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i k(\phi - y/N)} |y\rangle |0\rangle_a
\end{aligned} \tag{15}$$

Here, IQFT is applied to the c-register which is a superposition of basis states, $|k\rangle$. Using the distribution law of matrix operations, IQFT is applied to individual $|k\rangle$ and Eq. (V.87) is used with $y = j$ and $N = 2^n$.

QFT is the inverse of IQFT and can be treated as the rotation of the basis. The rotation matrix is given by

$$U_Q = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{(N-1)} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \omega^{(N-1)} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{V.88}$$

Equivalently,

$$U_Q |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} |j\rangle \tag{V.89}$$

It can be shown that $U_I = U_Q^{-1}$ or $U_I U_Q = I$.

12. Implementation of QFT and IQFT

QFT and IQFT are constructed using Hadamard gates, controlled phase shift gates, and SWAP gates. Readers may refer to other sources for the details (e.g. [18]). Here, we show

the circuit of a 2-qubit IQFT gate (Fig. 6).

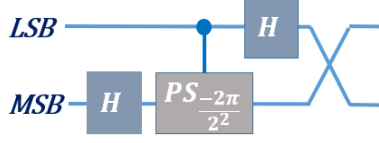


FIG. 6. Implementation of a 2-qubit Inverse Quantum Fourier Transformation

In general, the phase shift angle is $\phi = \frac{-2\pi}{2^r}$, $r - 1$ is the distance between the controlling qubit and the target qubit. For the 2-qubit IQFT case, there is only one controlled phase shift gate and $r = 2$ and this results in the phase $\phi = \frac{-\pi}{2}$.

For QFT, the circuit is the same as the IQFT, but the phase shift is negated, i.e. $\phi = \frac{2\pi}{2^r}$. This can be appreciated by the fact that the elements in the IQFT and QFT have opposition signs in Eq. (V.86) and Eq. (V.88), respectively.

13. Gaussian Elimination Method

Here, Gaussian Elimination is demonstrated by solving Eq. 1 using the numerical example in Section III.

$$A\vec{x} = \vec{b} \quad (1)$$

$$\begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (\text{V.90})$$

which is rewritten as an augmented matrix followed by Gaussian method of Elimination to solve for x_0 and x_1 ,

$$\begin{aligned} \begin{bmatrix} 1 & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 1 & 1 \end{bmatrix} &\xrightarrow{\text{Row2} = 3 \times \text{Row2} + \text{Row1}} \begin{bmatrix} 1 & -\frac{1}{3} & 0 \\ 0 & \frac{8}{3} & 3 \end{bmatrix} \xrightarrow{\text{Row2} = \frac{3}{8} \times \text{Row2}} \begin{bmatrix} 1 & -\frac{1}{3} & 0 \\ 0 & 1 & \frac{9}{8} \end{bmatrix} \\ &\xrightarrow{\text{Row1} = \frac{1}{3} \times (\text{Row1} + \text{Row2})} \begin{bmatrix} 1 & 0 & \frac{3}{8} \\ 0 & 1 & \frac{9}{8} \end{bmatrix} \end{aligned}$$

This the solution \vec{x} is

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} \frac{3}{8} \\ \frac{9}{8} \end{pmatrix}$$

The complexity of Gaussian Elimination is $O(n^3)$. This is much slower than the classical conjugate gradient method (Appendix V 14), to which HHL is compared.

14. Conjugate Gradient Method

The **Conjugate Gradient Method** (CGM) solves the LSP with a complexity of $O(N)$ and is the fastest known classical solver. Therefore, the speed of HHL, which has a complexity of $O(\log(N))$, is often compared to the speed of CGM [21]. Thus, HHL provides an exponential speedup over the fastest known classical method.

To solve,

$$A\vec{x} = \vec{b} \tag{1}$$

initial guess of \vec{x} is used as the starting point. The residual is then found and the search direction is determined by finding the steepest descent. This is repeated until a stable condition is met.

The residual in the k -th search is given as,

$$R_k = b - A\vec{x}_k \tag{V.91}$$

The readers do not need to understand CGM to understand HHL. Interested readers may refer to the literature (e.g. [9]) for more details.

-
- [1] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.
 - [2] C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. Deane, “The prospects of quantum computing in computational molecular biology,” WIREs Comput. Mol. Sci., **11**, e1481 (2021).
 - [3] D. J. Egger et al., “Quantum Computing for Finance: State-of-the-Art and Future Prospects,” in IEEE Transactions on Quantum Engineering, **1**, pp. 1–24, 2020, Art no. 3101724, doi: 10.1109/TQE.2020.3030314.

- [4] F. Arute, K. Arya, R. Babbush et al., “Quantum supremacy using a programmable superconducting processor,” *Nature* **574**, 505–510 (2019).<https://doi.org/10.1038/s41586-019-1666-5>.
- [5] National Strategic Overview for Quantum Information Science Report (National Science and Technology Council, 2018). <https://www.whitehouse.gov/wp-content/uploads/2018/09/National-Strategic-Overview-for-Quantum-Information-Science.pdf>.
- [6] N. David Mermin, “Could Feynman Have Said This?,” *Physics Today* **57** (5), 10 (2004); doi: 10.1063/1.1768652
- [7] A. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.* **103**, 150502 (2009).
- [8] Yudong Cao, Anmer Daskin, Steven Frankel, and Sabre Kais, “Quantum circuit design for solving linear systems of equations,” *Molecular Physics* **110**, 15–16 (2011).
- [9] R. Chandra, S.C. Eisenstat, and M.H. Schultz, ”Conjugate Gradient Methods for Partial Differential Equations,” in the Proceedings of the AICA International Symposium on Computer Methods for Partial Differential Equations, Bethlehem, Pennsylvania, June 1975.
- [10] S. Dmitry et al., “The Potential of Quantum Computing and Machine Learning to Advance Clinical Research and Change the Practice of Medicine.” *Missouri medicine* **115** (5), 463–467 (2018).
- [11] Bojia Duan, Jiabin Yuan, Chao-Hua Yu, Jianbang Huang, and Chang-Yu Hsieh, “A survey on HHL algorithm: From theory to application in quantum machine learning”, *Physics Letters A* **384**, 126595 (2020).
- [12] Shengbin Wang, Zhimin Wang, Wendong Li, Lixin Fan, Zhiqiang Wei, and Yongjian Gu, “Quantum fast Poisson solver: the algorithm and complete and modular circuit design,” *Quantum Information Processing* **19**, Article number: 170 (2020).
- [13] H. J. Morrell and H. Y. Wong, ”Study of using Quantum Computer to Solve Poisson Equation in Gate Insulators,” 2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), 2021, pp. 69-72, doi: 10.1109/SISPAD54002.2021.9592604.
- [14] Schleich, P., 2019. How to solve a linear system of equations using a quantum computer. http://www.acom.rwth-aachen.de/_media/3teaching/00projects/schleich.pdf.
- [15] HHL Example using Qiskit, <https://qiskit.org/textbook/ch-applications/hhl_tutorial.html>.

- [16] Gadi Aleksandrowicz, et al., (2019). Qiskit: An Open-source Framework for Quantum Computing (0.7.2). Zenodo. <https://doi.org/10.5281/zenodo.2562111>.
- [17] IBM Quantum Site, <<https://quantum-computing.ibm.com/>>.
- [18] Hiu Yung Wong, Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps. Switzerland: Springer Nature, 2022, pp. 170. <https://doi.org/10.1007/978-3-030-98339-0>. ISBN-10: 3030983382.
- [19] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Nairi Usher, and Leonard Wossnig, “Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Nairi Usher, Leonard Wossnig,” arXiv:1802.08227v1.
- [20] Matlab code and Jupyter Notebook, <https://github.com/hywong2/HHL_Example>.
- [21] Vandenbrocq, Adrien. ”On Quantum Algorithms for Solving Linear Systems of Equations.” Master’s semester Project I, 2019. <https://adrienvdb.com/projects-and-reports/>