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

ANIKA ZAMAN, HECTOR JOSE MORRELL, AND HIU YUNG WONG, (Senior Member, IEEE)
Department of Electrical Engineering, San Jose State University, San Jose, CA 95192, USA
Corresponding author: Hiu Yung Wong (hiuyung.wong@sjsu.edu)
This work was supported in part by the National Science Foundation under Grant 2046220.

ABSTRACT After learning basic quantum computing concepts, it is desirable to reinforce the learning using an important and relatively complex algorithm through which students can observe and appreciate how 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 basis 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 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.

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 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, a quantum breakthrough has been demonstrated using a 53-qubit system [4]. Quantum advantage has also been shown with a programmable photonic processor [5]. Therefore, the training of a quantum technology workforce is an imminent goal for many countries (e.g. [6]) 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 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 [7].

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 [8], [9] which can be used to solve linear system problems (LSP) and can provide exponential speedup over the classical conjugate gradient method [10] is chosen for this purpose. Although there are other algorithms which can be used to solve LSP (e.g. hybrid algorithm to solve LSP with limited qubit resources [11]), HHL is chosen for its simplicity. HHL is the basic of many more advanced algorithms and is important in various applications such as machine learning [12] and modeling of quantum systems [2], [13]. HHL solves system of linear equation which is a discretization of [14], [15]. 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. [16], [17]), 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 [18] corresponding to the numerical example is available and can be run in IBM simulation and hardware machines [19]. 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 operations, measurement, quantum Fourier transformation, and quantum programming (e.g. [21]). 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 the similar approach as in this paper can be found in [21].

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. LINEAR SYSTEM PROBLEM AND DEFINITIONS
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

\[ A\tilde{x} = \tilde{b} \] (1)

where \( A \) is a \( N_b \times N_b \) Hermitian matrix and \( \tilde{x} \) and \( \tilde{b} \) are \( N_b \)-dimensional vectors. \( A \) and \( \tilde{b} \) are known and \( \tilde{x} \) is the unknown to be solved. For simplicity, it is assumed \( N_b = 2^n_b \), where \( n_b \) is the number of qubits in the quantum circuit to solve the LSP. We can say that for \( n_b \) number of unknowns, we need \( n_b \) qubits to solve the LSP. Dummy equations can be added otherwise to convert the system to satisfy this assumption. The solution can be represented as,

\[ \tilde{x} = A^{-1}\tilde{b} \] (2)

As an example, \( A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \), \( \tilde{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \), and \( \tilde{x} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \) with \( n_b = 1 \) and \( N_b = 2^1 = 2 \). Readers may refer to Appendix V-M and Appendix V-N to review how LSP is solved classically using Gaussian Elimination and Conjugate Gradient Method, respectively.

\( A \) is assumed to be Hermitian (See Appendix V-A). If it is not Hermitian, then \( A \) can be converted to \( \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \), which is Hermitian. Readers may refer to [23] for the more advanced treatment when \( A \) is not Hermitian.

B. OVERVIEW OF HHL CIRCUIT
Figure 1 shows the schematic of the HHL algorithm and the corresponding circuit to solve LSP. In the HHL quantum algorithm, the \( N_b \) components of \( \tilde{b} \) and \( \tilde{x} \) are encoded as the amplitudes/coefficients (amplitude encoding) of basis states of the \( n_b \)-qubits, \( | \rangle_b \), which form a \( C^{N_b} \) Hilbert space. These \( n_b \) qubits are called b-register. Qubit \( n_b \) is chosen to be large enough to encode \( \tilde{b} \), i.e. \( 2^{n_b} \) needs to be the same as the length of the vectors \( \tilde{b} \) and \( \tilde{x} \). The matrix \( A \) is simulated through Hamiltonian encoding by encoding it as the Hamiltonian of a unitary gate. Appendix V-I 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⟩ in binary is |1⟩ in decimal because the 1 in the basis state |0001⟩ is the LSB, representing 2⁰ instead of 2³ (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 [18] and the IBM-Q platform [19].

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. The 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-H)), the value of n determines how accurately 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 \(|1⟩_a\) 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.

C. OVERVIEW OF HHL ALGORITHM

Here we will give an overview of the algorithm. The details will be discussed in the following subsections.

The matrix \( A \), which is a Hamiltonian, may be written as a linear combination of the outer products of its eigenvectors, \( |u_i⟩⟨u_i| \) weighted by its eigenvalues, \( \lambda_i \), in Eq. (3), (See Appendix V-H).

\[
A = \sum_{i=0}^{2^n-1} \lambda_i |u_i⟩⟨u_i|
\]  

(3)

Since \( A \) is diagonal in its eigenvector basis, its inverse is simply, \( A^{-1} = \sum_{i=0}^{2^n-1} \lambda_i^{-1} |u_i⟩⟨u_i| \). \( b \) can be also expressed in the basis formed by the eigenvectors of \( A \), such that

\[
|b⟩ = \sum_{j=0}^{2^n-1} b_j |u_j⟩
\]  

(4)

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

\[
|x⟩ = A^{-1} |b⟩ = \sum_{i=0}^{2^n-1} \lambda_i^{-1} b_i |u_i⟩
\]  

(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 c-register.

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

\[
\sum_{j=0}^{2^nb-1} |b_j|^2 = 1
\]

(6)

\[
\sum_{i=0}^{2^nb-1} |\lambda_i^{-1}b_i|^2 = 1
\]

(7)

To achieve the goal in Eq. (5), the first step is to encode \(|b\rangle\) in the b-register through amplitude encoding and encode \(A\) in the controlled unitary gate \((C - U)\) in QPE as \(U = e^{iA}\), through Hamiltonian encoding.

The purpose of QPE (Fig. 1) is to find the eigenphase of \(U\) which is just the eigenvalue of \(A\) and store it in the c-register. This is achieved through superposition creation (the Hadamard gates), controlled rotations \((C - U\) gates) and Inverse Quantum Fourier Transform (IQFT), which performs constructive and destructive interference so that only the basis corresponding to the eigenvalue will have non-zero amplitude and is stored in the c-register (basis encoding).

The reason to store the eigenvalue of \(A\) in the c-register is to prepare to perform another controlled rotation based on the eigenvalue on the ancilla qubit. It is carefully designed so that after this rotation, \(\lambda_i^{-1}b_i\) (as required by Eq. (5)) will appear in the state of the quantum circuit after measurement on \(|\rangle\rangle\). The measurement outcome is probabilistic. If \(|\rangle\rangle\rangle\rangle\) is obtained, the result will be discarded and the circuit needs to be rerun. If \(|\rangle\rangle\rangle\rangle\) is obtained, then \(\lambda_i^{-1}b_i\) will appear in the coefficients of the quantum state of the circuit.

After the controlled rotation on the ancilla qubit, in principle, the solution has been encoded in the state of the circuit. However, the b-register and c-register are entangled. Interested readers are suggested to read the explanation in Appendix V-F on why the entanglement prevents us from obtaining the correct answer. Therefore, an uncomputation using inverse QPE is required to disentangle the two registers and have the answer stored in the b-register as in Eq. (5).

**D. NATURES OF THE HHL ALGORITHM**

It will be instructive to note a few natures and limitations of the HHL algorithm before the readers diving into the details.

Firstly, HHL algorithm provides an exponential speed-up over conjugated gradient only for sparse matrix, \(A\). Fortunately, many scientific problems such as computational science in engineering have sparse matrices (E.g. [15]).

While HHL provides an execution complexity of \(\mathcal{O}(\log(N))\), the readout of the result is \(\mathcal{O}(N)\). This renders HHL useless if reading each component of \(|x\rangle\) is required. However, this algorithm can be combined with others in situations such as finding the expectation value of \(|x\rangle\) of an operator \(M\) (i.e. \(\langle x|M|x\rangle\)).

As it will be clear later, the number of qubits in the c-register determines the accuracy of the eigenvalues of \(A\) (which is the very nature of the “estimation” in QPE) and thus the solution. The complexity of the algorithm actually depends on \(k\log(N)/\epsilon\), where \(k\) is the system’s condition, \(s\) is the sparsity, and \(\epsilon\) is the error desired. Therefore, readers should be aware of this before making a judgment if HHL is useful for their problems.

As will be seen later, efficient encoding of \(U\) and construction of controlled-ancilla bit rotation are not trivial (e.g. [14]). Readers need to avoid more than \(\mathcal{O}(\log(N))\) complexity in their construction time.

Finally, in some problems, HHL is shown to have exponential speed-up over existing classical algorithms but there is still a possibility that new classical algorithms can achieve the same speed-up. Readers are suggested to read the paper by Aaronson to understand more [22].

**E. STATE PREPARATION**

There are total \(n_b + n + 1\) qubits and they are initialized as

\[
|\Psi_0\rangle = |0\cdots0\rangle_b|0\cdots0\rangle_\ast|0\rangle_a = |0\rangle^{\otimes n_b}|0\rangle^{\otimes n}|0\rangle
\]

(8)

In the state preparation, \(|0\cdots0\rangle_b\) in the b-register needs to be rotated to have the amplitudes correspond to the coefficients of \(\tilde{b}\). That is

\[
\tilde{b} = \begin{pmatrix}
\beta_0 \\
\beta_1 \\
\vdots \\
\beta_{n_b-1}
\end{pmatrix} \Leftrightarrow \beta_0|0\rangle + \beta_1|1\rangle + \cdots + \beta_{n_b-1}|n_b-1\rangle = |b\rangle
\]

(9)

The vector \(\tilde{b}\) is represented in a column form on the left with coefficients \(\beta_i\’s\). 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,

\[
|\Psi_1\rangle = |b\rangle_b|0\cdots0\rangle_\ast|0\rangle_a
\]

(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 \(\tilde{b}\), it will be discussed in more detail in the numerical example.

**F. 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 \(n_b\) eigenvalues of the unitary rotation matrix, \(U = e^{iA}\), 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 roots of 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 Appendices V-K and V-L.

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

\[
|\Psi_2\rangle = H^{\otimes n} \otimes H^{\otimes n} \otimes I |\Psi_1\rangle = |b\rangle \frac{1}{2^n} (|0\rangle + |1\rangle)^{\otimes n} |0\rangle
\]

\[
|\Psi_2\rangle = \frac{1}{2^n} (|0\rangle + |1\rangle)^{\otimes n} |0\rangle
\]

In the IQFT part, Eq. (15), only the clock qubits are affected. Note that in certain literature, this is called Quantum Fourier Transform (QFT) (Appendix V-K).

\[
|\Psi_4\rangle = |b\rangle \text{IQFT}(\frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i b k} |k\rangle)0_a
\]

\[
= |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i b k} (\text{IQFT}|k\rangle)0_a = |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i b k} (\sum_{y=0}^{2^n-1} e^{-2\pi i y/N} |y\rangle)0_a
\]

\[
= \frac{1}{2^n} |b\rangle \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i k(y/N - \phi)} |y\rangle0_a
\]

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

\[
|\Psi_4\rangle = \frac{1}{2^n} |b\rangle \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i k(y/N - \phi)} |y\rangle0_a = |b\rangle |N\phi\rangle0_a
\]

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

To understand why the \( c \)-register is called the clock qubits and the meaning of \( t \) better, it is worth noting that the Hamiltonian of a system in quantum mechanics determines how a system evolves through the following equation,

\[
U = e^{iAt}
\]

Therefore, \( t \) can be treated as the evolution time for that Hamiltonian.

\( U \) is also diagonal in \( A \)'s eigenvector, \(|u_j\rangle\), basis. If \(|b\rangle = |u_j\rangle\),

\[
U|b\rangle = e^{i\lambda_j t} |u_j\rangle
\]

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

\[
|\Psi_4\rangle = |u_j\rangle |N\lambda_j t/2\pi\rangle0_a
\]

Thus the eigenvalues of \( A \) have been encoded in the clock qubits (basis encoding).

So far, we have assumed that \(|b\rangle \) is an eigenvector of \( U \), \(|u_j\rangle\). In general, \(|b\rangle \) can be expressed as a superposition of \(|u_j\rangle \) (Eq. 4) and by superposition,

\[
|\Psi_4\rangle = \sum_{j=0}^{2^n-1} b_j |u_j\rangle |N\lambda_j t/2\pi\rangle0_a
\]

The \( \lambda_j \) are usually not integers. We will choose \( t \) so that \( \lambda_j = N \lambda_j t/2\pi \) are integers. Therefore, \( \lambda_j \) are usually scaled version of \( \lambda_j \).
\[ |\Psi_4\rangle = \sum_{j=0}^{2^n-1} b_j|u_j\rangle|\tilde{\lambda}_j\rangle|0\rangle_a \]  

### G. 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,

\[ |\Psi_5\rangle = \sum_{j=0}^{2^n-1} b_j|u_j\rangle|\tilde{\lambda}_j\rangle(1 - \frac{C^2}{\lambda_j^2}|0\rangle_a + \frac{C}{\lambda_j}|1\rangle_a) \]

where \(C\) is a constant. The following 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

\[ |\Psi_6\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle|\tilde{\lambda}_j\rangle^2}} \sum_{j=0}^{2^n-1} b_j|u_j\rangle|\tilde{\lambda}_j\rangle C|1\rangle_a \]

where the prefactor is due to normalization after measurement. Since \(|\frac{C}{\lambda_j}\rangle^2\) is proportional to the probability of obtaining \(|1\rangle\) when the ancilla bit is measured, \(C\) should be chosen to be as large as possible. Compared to (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 \(|0\rangle/|1\rangle\)). Moreover, 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. Interested readers are suggested to read the explanation in Appendix V-F on why the entanglement prevents us from obtaining the correct answer. 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.

### H. UNCOMPUTATION - INVERSE QPE

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

\[ |\Psi_7\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle^2|QFT|\tilde{\lambda}_j\rangle^2}} \sum_{j=0}^{2^n-1} b_j|u_j\rangle|QFT|\tilde{\lambda}_j\rangle|1\rangle_a \]

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 \(|1\rangle\), \((U^{-1})^2\) will be applied to \(|u_j\rangle\). This is equivalent to multiplying \(e^{-i\lambda_j}\) if the c-register is \(|y\rangle\). This is due to the similar argument in Eq. (14) and the fact that \(2\pi i\phi = i\lambda_j\). Therefore,

\[ |\Psi_8\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle^2}} \sum_{j=0}^{2^n-1} b_j|u_j\rangle \sum_{j=0}^{2^n-1} \exp(-i\lambda_j|y\rangle) |1\rangle_a \]

Since we earlier chose to set \(\tilde{\lambda}_j = N\lambda_j/2\pi\), therefore, the two exponential terms cancel each other and

\[ |\Psi_8\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle^2}} \sum_{j=0}^{2^n-1} b_j|u_j\rangle \sum_{j=0}^{2^n-1} \exp(-i\lambda_j|y\rangle) |1\rangle_a \]

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

\[ |\Psi_9\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle^2}} \sum_{j=0}^{2^n-1} b_j|u_j\rangle |0\rangle_a \otimes |1\rangle_a \]

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

\[ |\Psi_9\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} b_j|u_j\rangle^2}} |x\rangle |b\rangle |0\rangle_a \otimes |1\rangle_a \]

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

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}$$  \hspace{1cm} (29)

$$\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$  \hspace{1cm} (30)

The eigenvectors of $A$ are $\vec{u}_0 = \left( \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right)$, $\vec{u}_1 = \left( \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right)$ with eigenvalues $\lambda_0 = \frac{2}{3}$ and $\lambda_1 = \frac{1}{3}$ respectively. We need to using basis encoding to encode the eigenvalues in the basis formed by the clock qubit and 2 qubits are needed by encoding $\lambda_0$ as $|01\rangle$ and $\lambda_1$ as $|10\rangle$ so that it maintains the ratio of $\lambda_1/\lambda_0 = 2$. This means $\lambda_0 = 1$ and $\lambda_1 = 2$ or in other words, $|\lambda_0\rangle = |01\rangle$ and $|\lambda_1\rangle = |10\rangle$. This gives a perfect encoding with $n = 2$ (i.e. $N = 4$). Therefore, $i$ is chosen to be $\frac{3\pi}{4}$ to achieve the encoding scheme since $\lambda_j = N\lambda_j/2\pi$.

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

$$\vec{x} = \left( \begin{pmatrix} \frac{3}{8} \\ \frac{5}{8} \end{pmatrix} \right)$$  \hspace{1cm} (31)

whereby, the ratio of $|\lambda_0|^2$ to $|\lambda_1|^2$ is $1 : 9$.

Here, we have applied a NOT gate to encode $\vec{b}$ from the initial ground state. More complex gates are needed for more general or higher dimensional $\vec{b}$. For example, a Hadamard gate can be used if $\vec{b} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

B. CONTROLLED-U IMPLEMENTATION

In reality, we expect the controlled-$U$ operation to be implemented by Hamiltonian simulation [20]. However, to understand the algorithm, we will derive the matrix for $U$ and then map this to the Controlled − $U$ gate used in IBM-Q directly. Since $n = 2$, there are two operations needed, namely $U^{2\pi}_t = U^2$ and $U^{2\pi} = U$, 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 transforms back to the original basis.

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

$$V = (\vec{u}_0 \vec{u}_1)$$

$$= \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$  \hspace{1cm} (32)

Since $V$ is real and symmetric, its conjugate, $V^\dagger$ equals itself.

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

$$A_{\text{diag}} = V^\dagger AV$$

This is expected as every matrix expressed in its eigenvector basis must be diagonal with its eigenvalues along the diagonals. And since it is diagonal, $U$ can be obtained by the exponentiation of the elements accordingly.

$$U_{\text{diag}} = \begin{pmatrix} e^{i\lambda_0 t} & 0 \\ 0 & e^{i\lambda_1 t} \end{pmatrix}$$

$$= \begin{pmatrix} e^{i\pi/2} & 0 \\ 0 & e^{i\pi} \end{pmatrix}$$

$$= \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix}$$  \hspace{1cm} (34)

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

$$U^2_{\text{diag}} = U_{\text{diag}} U_{\text{diag}} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$  \hspace{1cm} (35)

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, after

$$U = VU_{\text{diag}} V^\dagger$$

$$= \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 1 & 1 + i \\ 1 + i & 1 \end{pmatrix}$$  \hspace{1cm} (36)

$$U^2 = VU^2_{\text{diag}} V^\dagger$$

$$= \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$  \hspace{1cm} (37)

Up to here, we have demonstrated how to find the $U$’s in their matrix forms. We then need to map them to the rotations on Bloch Sphere so that it can be implemented by more fundamental gates.

To implement $U$ and $U^2$, a 4-parameter arbitrary unitary gate with global phase can be used [21],

$$U = \begin{pmatrix} e^{i\gamma \cos(\theta/2)} & -e^{i(\gamma + \lambda) \sin(\theta/2)} \\ e^{i(\gamma + \phi) \sin(\theta/2)} & e^{i(\gamma + \phi + \lambda) \cos(\theta/2)} \end{pmatrix}$$  \hspace{1cm} (38)

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

By choosing $\theta = \pi/2$, $\phi = -\pi/2$, $\lambda = 0$, $\gamma = 3\pi/4$, $U$ 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,

$$U^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$$  \hspace{1cm} (39)
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 \(#A#\)

\[
C - U' = I \otimes |0\rangle |0\rangle + U' \otimes |1\rangle |1\rangle
\]  

(40)

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

\[
\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1 & i & -1 & i \\ 1 & 1 & 1 & 1 \end{pmatrix} \]

\[
= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & i & 0 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & i & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}
\]

\[
\approx \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & i & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & i & 0 \end{pmatrix}
\]  

(41)

At this step, we have implemented the C-U gates in QPE and IQPE using arbitrary unitary gates which is available in common quantum computing platform such as IBM-Q (Figure 3). The same methodology can be used for higher-dimensional problems.

**C. IMPLEMENTATION 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 \( \frac{1}{\sqrt{\lambda_i}} \) and \( \frac{c}{\sqrt{\lambda_i}} \), respectively. The sum of the square of the magnitudes of the coefficients is 1 as required. This means also \( C \leq \lambda_i \). Since the minimal \( \lambda_i \) is 1, we will set \( C = 1 \) to maximize the probability of measuring \( |1\rangle \) during the ancilla bit measurement.

The transformation of \( |0\rangle \) to \( \frac{1}{\sqrt{\lambda_i}} |0\rangle + \frac{1}{\sqrt{\lambda_i}} |1\rangle \) is known to be equivalent to \( RY(\theta) \) rotation,

\[
RY(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}
\]  

(42)

with \( \theta = 2\arcsin(\frac{1}{\sqrt{\lambda_i}}) \). One can check this by multiplying \( RY(\theta) \) to \( |0\rangle \) as follows,

\[
RY(\theta) |0\rangle = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \cos(\theta/2) |0\rangle + \sin(\theta/2) |1\rangle
\]  

(43)

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 (21). The function is defined as

\[
\theta(c) = \theta(c_1 c_0) = 2 \arcsin(\frac{1}{c})
\]  

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

Since only \( |\lambda_i\rangle \) has non-zero amplitude in (21), we only need to set up (44) such that it is correct for \( |c\rangle = |01\rangle \) and \( |10\rangle \), namely

\[
\theta(1) = \theta(01) = 2 \arcsin(\frac{1}{2}) = \pi
\]  

(45)

\[
\theta(2) = \theta(10) = 2 \arcsin(\frac{1}{2}) = \frac{\pi}{3}
\]  

(46)

The following function can achieve the goal,

\[
\theta(c) = \theta(c_1 c_0) = \frac{\pi}{3} c_1 + \pi c_0
\]  

(47)

Therefore, the controlled rotation can be implemented as

\[
CU3 = |1\rangle |1\rangle \otimes I \otimes RY\left(\frac{\pi}{3}\right) + |0\rangle |0\rangle \otimes I \otimes I
\]

\[
\quad + I \otimes |1\rangle |1\rangle \otimes RY(\pi) + I \otimes |0\rangle |0\rangle \otimes I
\]  

(48)

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

It should be noted that, for didactic purposes, the controlled rotation gates are deduced with the knowledge of the eigenvalues of \( A \), which are unknown in real applications. Users can refer to the literatures for some more advanced and systematic methods (e.g. [14]).

**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

\[
|\Psi_0\rangle = |0\rangle_b \otimes |00\rangle_c \otimes |0\rangle_a = |0000\rangle
\]  

(49)

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

\[
|\Psi_1\rangle = X \otimes I \otimes |\Psi_0\rangle = |1000\rangle
\]  

(50)

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

\[
|\Psi_2\rangle = I \otimes H^{\otimes a} \otimes I |\Psi_1\rangle
\]

\[
= |1\rangle \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)^{\otimes 2} |0\rangle
\]

\[
= \frac{1}{\sqrt{2}} (|1000\rangle + |0100\rangle + |1100\rangle + |1110\rangle)
\]  

(51)

It is clear that the superposition is achieved due to the multiplication of \( (|0\rangle + |1\rangle) \) among each others.

Before applying the CU3 (controlled rotation of ancillary qubit) gates in the bra-ket notation, it will be convenient to perform a basis change to the eigenvector basis of \( A \). Since \( |1\rangle = \frac{1}{\sqrt{2}} (|\alpha_0\rangle + |\alpha_1\rangle) \), we have \( b_0 = \frac{1}{\sqrt{2}} \) and \( b_1 = \frac{1}{\sqrt{2}} \).

Therefore,

\[
|\Psi_2\rangle = |1\rangle \frac{1}{\sqrt{2}} (|0000\rangle + |0100\rangle + |1000\rangle + |1100\rangle)
\]
With $|\Psi_2\rangle$ expressed as the linear combination of the eigenvectors of $A$, i.e. $|u_i\rangle$, we will be able to extract the eigenphases of $U$ easily.

In the controlled rotation operations, when the corresponding c-register is $|k\rangle_c$, a phase change of $\phi_j = k \lambda_j / 2\pi$ is added (i.e. multiplied by $e^{2\pi i \phi_j}$) for $|u_j\rangle$. Since $t = \frac{3\pi}{4}$, $\lambda_0 = \frac{3}{4}$ and $\lambda_1 = \frac{3}{4}$, we have

$$|\Psi_3\rangle = \frac{1}{2\sqrt{2}}(-|u_0\rangle|000\rangle + e^{2\pi i \phi_0}|u_0\rangle|010\rangle + |u_0\rangle|100\rangle$$

$$|\Psi_3\rangle = \frac{1}{2\sqrt{2}}(-|u_0\rangle|000\rangle - |u_0\rangle|010\rangle - |u_0\rangle|100\rangle)$$

$$|\Psi_3\rangle = \frac{1}{2\sqrt{2}}(-|u_0\rangle|000\rangle - i|u_0\rangle|010\rangle + |u_0\rangle|100\rangle)$$

Before applying IQFT, the terms are regrouped for simplicity.

$$|\Psi_5\rangle = \frac{1}{2\sqrt{2}}((-|u_0\rangle + |u_1\rangle)|000\rangle + (-i|u_0\rangle - |u_1\rangle)|010\rangle + (|u_0\rangle + |u_1\rangle)|100\rangle)$$

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

$$\text{IQFT}|10\rangle = \text{IQFT}|2\rangle$$

$$= \frac{1}{2^{2/2}} \sum_{y=0}^{2^2-1} e^{-2\pi i y/4} |y\rangle$$

$$= \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle)$$

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

Similarly,

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

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

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

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

$$|\Psi_4\rangle = \text{IQFT}|\Psi_3\rangle$$

$$= \frac{1}{4\sqrt{2}}((-|u_0\rangle + |u_1\rangle)(|00\rangle + |01\rangle + |10\rangle + |11\rangle))$$
+ (-i|u_0| - |u_1|)(|00⟩ - |01⟩ + |10⟩ + i|11⟩)
+ (|u_0| + |u_1|)(|00⟩ - |01⟩ + |10⟩ - |11⟩)
+ (i|u_0| - |u_1|)(|00⟩ + |01⟩ - |10⟩ - i|11⟩)|0⟩
= 1/\sqrt{2}(-|u_0⟩|01⟩ + |u_1⟩|10⟩)|0⟩
(59)

It can be seen that after IQFT, the eigenvalues are encoded in the clock qubits as |01⟩ and |11⟩ with non-zero amplitudes due to constructive interference. The readers are encouraged to read it carefully to see how the coefficients of |00⟩ and |11⟩ terms cancel each other (destructive 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⟩ goes with |01⟩ and |u_1⟩ goes with |11⟩.

After performing the ancilla qubit rotation,

|Ψ_5⟩ = \sum_{j=0}^{2^{l-1}} b_j|u_j⟩|\tilde{λ}_j⟩(\sqrt{1 - C^2/\tilde{λ}_j}^2|0⟩ + C/\tilde{λ}_j|1⟩)
= -\frac{1}{\sqrt{2}}|u_0⟩|0⟩(\sqrt{1 - \frac{1}{12}|0⟩ + \frac{1}{1}|1⟩)
+ \frac{1}{\sqrt{2}}|u_1⟩|10⟩(\sqrt{1 - \frac{1}{22}|0⟩ + \frac{1}{2}|1⟩)
(60)

If the measurement of the ancilla bit is |1⟩,

|Ψ_6⟩ = \sqrt{\frac{8}{5}}\left(-\frac{1}{\sqrt{2}}|u_0⟩|01⟩|1⟩
+ \frac{1}{2\sqrt{2}}|u_1⟩|10⟩|1⟩\right)
(61)

Applying QFT to the encoded eigenvalues, we have

QFT|10⟩ = QFT|2⟩
= \frac{1}{2\sqrt{2}} \sum_{y=0}^{2^{l-1}} e^{2\pi i/2^l y/4}|y⟩
(62)

QFT|01⟩ = QFT|1⟩
= \frac{1}{2}(|00⟩ + i|01⟩ - |10⟩ - i|11⟩)
(63)

Therefore, applying QFT to |Ψ_6⟩ and substituting Eq. (62) to (63), we obtain

|Ψ_7⟩ = \sqrt{\frac{8}{5}}(-\frac{1}{\sqrt{2}}|u_0⟩\frac{1}{2}(|00⟩ + i|01⟩ - |10⟩
- i|11⟩)|1⟩ + \frac{1}{2\sqrt{2}}|u_1⟩\frac{1}{2}(|00⟩ - |01⟩
+ |10⟩ - |11⟩)|1⟩
(64)

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

|Ψ_8⟩ = \sqrt{\frac{8}{5}}\left(-\frac{1}{\sqrt{2}}|u_0⟩\frac{1}{2}(|00⟩ + |01⟩ + |10⟩ + |11⟩)|1⟩
+ \frac{1}{2\sqrt{2}}|u_1⟩\frac{1}{2}(|00⟩ + |01⟩ + |10⟩ + |11⟩)|1⟩
+ \frac{1}{2\sqrt{2}}|u_1⟩\frac{1}{2}(|00⟩ + |01⟩ + |10⟩ + |11⟩)|1⟩
+ \frac{1}{2\sqrt{2}}|u_1⟩\frac{1}{2}(|00⟩ + |01⟩ + |10⟩ + |11⟩)|1⟩
= \frac{1}{2\sqrt{2}}|u_1⟩\frac{1}{2}(|00⟩ + |01⟩ + |10⟩ + |11⟩)|1⟩
(65)

Finally, by applying Hadamard gate to the clock qubits,

|Ψ_9⟩ = \frac{2}{3}\sqrt{\frac{8}{5}}\left(-\frac{1}{\sqrt{2}}|u_0⟩ + \frac{1}{\sqrt{2}}|u_1⟩\right)|0⟩|1⟩
(66)

It can be verified that |Ψ_9⟩ is a normalized vector as it should be because every operation in the HHL circuit is unitary and preserves the norm.

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

|Ψ_9⟩ = \frac{1}{2\sqrt{5}}(|0⟩ + 3|1⟩)|0⟩|1⟩
(67)

The probability ratio of obtaining |0⟩ and |1⟩ 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 [24]. In the Matlab code, measurement is not performed (i.e. not partial tracing of the matrix). |Ψ_0⟩ is found to be,

\begin{bmatrix}
-0.4330 \\
0.2500 \\
0.0000 \\
-0.0000 \\
0.0000 \\
0.0000 \\
0.4330 \\
0.7500 \\
-0.0000 \\
0.0000 \\
-0.0000 \\
0.0000 \\
-0.0000 \\
0.0000 \\
0.0000
\end{bmatrix}
\begin{bmatrix}
|ψ_9⟩
\end{bmatrix}
(68)

Since |0⟩_c are discarded during the measurement step, only |0001⟩ and |1001⟩ 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 [24]). 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⟩_a should be considered. The ratio of the measurement probability of |0⟩_b|1⟩_a...
to $|1\rangle_b |1\rangle_a$ is 0.063 : 0.564 = 1 : 8.95, which is close to the expected value.

![Graph](image)

**FIGURE 4.** Simulation result of the circuit in Figure 3 using IBM – Q. Only the MSB $|b\rangle$ and the LSB $|a\rangle$ are measured.

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 $0.142^2 : 0.361^2 = 1 : 2.54$.

![Graph](image)

**FIGURE 5.** Hardware result of the circuit in Figure 3 run in machine ibmq_santiago. Only the MSB $|b\rangle$ and the LSB $|a\rangle$ are measured.

## 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 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

### A. 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,

$$A = A^\dagger = (A^T)^*$$

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. 32), where $V$ is Hermitian.

$$V = \begin{pmatrix} u_0 & u_1 \\ \bar{u}_1 & \bar{u}_0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

### B. 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 $\langle v| = (1 \ i)$.

### C. 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 $|v\rangle = c_1 |0\rangle + c_2 |1\rangle$, where $c_1$ and $c_2$ are complex number and $|0\rangle$ and $|1\rangle$ are basis states. A Hadamard gate is a gate commonly used to create a superposition state (Appendix V-E).

### D. 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 vector(s). Usually, a quantum gate rotates a basis state into another basis state and has its classical counterpart (e.g. the NOT gate). 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-E).

### E. 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 $|1\rangle$. 

---

**VOLUME 11, 2023**

77127

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The matrix form of the Hadamard gate is,
\[
\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\]

(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,
\[
\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}
\]

(V.71)

which can also be represented in bra-ket form as,
\[
\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)
\]

(V.72)

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),
\[
|\Psi_2\rangle = I^\otimes n_b \otimes H^\otimes n \otimes I|\Psi_1\rangle
\]

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.

**F. 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,
\[
|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\]

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

(V.75)

where the b-register and the c-register are entangled and \(|u_0\rangle \langle u_0|\) always appears with \(|01\rangle\langle 1|\) after the measurement.

If the b-register were not entangled with the c-register, we have
\[
|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left( -\frac{1}{\sqrt{2}} |u_0\rangle |0\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle |1\rangle \right)
\]

By substituting \(|u_0\rangle = -\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle\) and \(|u_1\rangle = \frac{1}{2} |0\rangle + \frac{1}{2\sqrt{2}} |1\rangle\) and after simplification, we have
\[
|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left( -\frac{1}{\sqrt{2}} \left( -\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) + \frac{1}{2\sqrt{2}} \left( -\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \right)
\]

(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.
\[
|\Psi_6\rangle = \sqrt{\frac{8}{5}} \left( -\frac{1}{\sqrt{2}} \left( -\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) |01\rangle \right) + \frac{1}{2\sqrt{2}} \left( -\frac{1}{\sqrt{2}} |0\rangle \right) + \frac{1}{\sqrt{2}} |1\rangle |10\rangle
\]

(V.77)

**G. 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 Fig. 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. \(|bc_{n-1} \cdots \rangle = |01 \cdots \rangle\).

In general, the controlled version of a unitary gate, \(U'\), can be implemented using the following equation if the LSB is the controlling qubit.
\[
C - U' = I \otimes |0\rangle \langle 0| + U' \otimes |1\rangle \langle 1|
\]

(V.78)

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

**H. EIGENVALUE AND EIGENVECTOR**

When a non-zero \(n \times n\) matrix \(A\) is applied to an \(n\)-dimensional vector \(\tilde{V}\) with the following relationship,
\[
A\tilde{V} = \lambda \tilde{V}
\]

(V.79)

where \(\lambda\) is a scalar, then, by definition, \(\tilde{V}\) and \(\lambda\) are the eigenvector and eigenvalue of \(A\), respectively. This is similar to Eq. (3), where the matrix \(A\) is expressed as a linear combination of the outer products of its eigenvectors, \(|u_i\rangle\langle u_i|\).
\[
A = \sum_{i=0}^{2^n-1} \lambda_i |u_i\rangle \langle u_i|
\]

(V.80)

This can be checked by applying \(A\) to its eigenvector \(|u_i\rangle\),
\[
A|u_j\rangle = \sum_{i=0}^{2^n-1} \lambda_i |u_i\rangle \langle u_i|u_j\rangle
\]

(V.81)
\[
= \sum_{i=0}^{2^n-1} \lambda_i |u_i\rangle \delta_{ij}
\]

(V.82)
\[
= \lambda_j |u_j\rangle
\]

(V.83)

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

\[
x = 2 \rightarrow 10 \text{ quantum state } |10\rangle
\]

\[
x = \left(\frac{2}{3}\right) \text{ binary } \left(\begin{array}{l}10 \\ 11\end{array}\right) \text{ quantum state } |1011\rangle \quad (V.80)
\]

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

\[
\vec{v} = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}
\]

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

\[
|v⟩ = v_0|0⟩ + v_1|1⟩ \quad (V.82)
\]

where v₀ and v₁ become the coefficients of the basis states, |0⟩ and |1⟩, respectively. In the main text, Eq. (9) shows how the values of the components of vector |b⟩ 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

\[
U = e^{iAt}
\]

where it encodes matrix A as the Hamiltonian of the unitary gate U. Matrix A needs to be Hermitian as it is used to represent the Hamiltonian (the energy) of the system. However, A does not need to be unitary and U will be unitary due to its definition in Eq. (17).

J. DISCRETE FOURIER TRANSFORM (DFT)

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

\[
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{N-1}
\end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix}
\omega^{0 \cdot 0} & \cdots & \omega^{0 \cdot (N-1)} \\
\omega^{-1 \cdot 0} & \cdots & \omega^{-1 \cdot (N-1)} \\
\vdots & \ddots & \vdots \\
\omega^{-(N-1) \cdot 0} & \cdots & \omega^{-(N-1) \cdot (N-1)}
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_1 \\
\vdots \\
x_{N-1}
\end{pmatrix}
\]  

(V.83)

K. INVERSE QUANTUM FOURIER TRANSFORM (IQFT) AND QUANTUM FOURIER TRANSFORM (QFT)

Mathematically, IQFT is similar to DFT (See Appendix V-J). 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.83) becomes

\[
|Y⟩ = |U_I|X⟩ \quad (V.84)
\]

and \( U_I \) has the same expression as \( \Omega \) in DFT.

\[
U_I = \frac{1}{\sqrt{N}} \begin{pmatrix}
\omega^{0 \cdot 0} & \cdots & \omega^{0 \cdot (N-1)} \\
\omega^{-1 \cdot 0} & \cdots & \omega^{-1 \cdot (N-1)} \\
\vdots & \ddots & \vdots \\
\omega^{-(N-1) \cdot 0} & \cdots & \omega^{-(N-1) \cdot (N-1)}
\end{pmatrix} \quad (V.85)
\]

Note that in some literature, e.g. [21], this form of IQFT is called QFT. \(|X⟩\) and \(|Y⟩\) are the quantum states/vectors in the N-dimensional Hilbert space. IQFT can be treated as the rotation of \(|X⟩\) to \(|Y⟩\).

If \(|X⟩\) is a basis vector \(|k⟩\), by applying IQFT to \(|k⟩\) using Eq. (V.85), we have

\[
U_I|k⟩ = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{-jk}|j⟩ \quad (V.86)
\]

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,

\[
\text{IQFT}|10⟩ = \text{IQFT}|2⟩
\]

\[
= \frac{1}{\sqrt{2}^2} \sum_{j=0}^{2^2-1} e^{-2\pi i j/4}|y⟩
\]

\[
= 1 \cdot (|00⟩ - |01⟩ + |10⟩ - |11⟩) \quad (55)
\]

where \( N = 2^2 = 4 \), \( k = 2, j = y \) in (V.86) is used. The basis state \(|10⟩\) becomes a superposition of all other basis states after IQFT.

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

\[
|\Psiₜ⟩ = |b⟩\text{IQFT}\left(\frac{1}{\sqrt{2}^2} \sum_{k=0}^{2^2-1} e^{2\pi i Φk}|k⟩\right)|0⟩_a
\]

\[
= |b⟩ \frac{1}{\sqrt{2}^2} \sum_{k=0}^{2^2-1} e^{2\pi i Φk} (\text{IQFT}|k⟩)|0⟩_a
\]

\[
= |b⟩ \frac{1}{\sqrt{2}^2} \sum_{k=0}^{2^2-1} e^{2\pi i Φk} \left(\sum_{y=0}^{2^2-1} e^{-2\pi i yk/N}|y⟩\right)|0⟩_a
\]

\[
= \frac{1}{\sqrt{2}^2} |b⟩ \sum_{y=0}^{2^2-1} \sum_{k=0}^{2^2-1} e^{2\pi i k(Φ - y/N)}|y⟩|0⟩_a \quad (15)
\]
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 (V.86) 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

$$ U_Q = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \cdots & \omega^{(N-1)} \\ 1 & \omega^2 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \quad \text{(V.87)} $$

Equivalently,

$$ U_Q |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} |j\rangle \quad \text{(V.88)} $$

It can be shown that $U_1 = U_Q^{-1}$ or $U_1 U_Q = I$.

**L. 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. [21]). Here, we show the circuit of a 2-qubit $IQFT$ gate (Fig. 6).

**FIGURE 6.** Implementation of a 2-qubit inverse quantum Fourier transformation.

In general, the phase shift angle is $\phi = \frac{2\pi}{N}$, $r - 1$ 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 $\phi = \frac{\pi}{4}$.

For $QFT$, the circuit is the same as the $IQFT$, but the phase shift is negated, i.e. $\phi = \frac{-\pi}{2}$. This can be appreciated by the fact that the elements in the $IQFT$ and $QFT$ have opposition signs in (V.85) and (V.87), respectively.

**M. GAUSSIAN ELIMINATION METHOD**

Here, Gaussian elimination is demonstrated by solving (1) using the numerical example in Section III.

$$ Ax = b \quad \text{(1)} $$

which is rewritten as an augmented matrix followed by the Gaussian method of elimination to solve for $x_0$ and $x_1$.

$$ \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{9} \\ \frac{1}{3} & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{(V.89)} $$

This solution $\bar{x}$ is

$$ \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad \text{(V.90)} $$

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

**N. CONJUGATE GRADIENT METHOD**

The Conjugate Gradient Method (CGM) solves the LSP with a complexity of $O(N)$ and is the fastest known classical solver. HHL, which has a complexity of $O(\log(N))$, is often compared to the speed of CGM [25]. Thus, HHL provides an exponential speedup over the fastest known classical method.

In LSP, according to Eq. (1), $A$ is a matrix, $b$ is a vector and $x$ is the variable to be solved. If $A$ is a $2 \times 2$ matrix and $b$ is $2 \times 1$, then $x$ can be solved easily like the example in this paper. But if $A$ is a large matrix, for example, $10^9 \times 10^9$ and $b$ is $10^9 \times 1$ vector and $N$, in this case, is $10^9$, the time will be in the order of $10^{27}$ units if we will solve it with the classical Gaussian Elimination method ($O(N^3)$). Using the classical Conjugate Gradient Method and if $A$ is sparse, it will take the order of $10^9$ units of time. And for HHL Quantum Algorithm, for a sparse matrix, $A$, it takes only $O(\log(N)) \approx 9$ units of time (ignoring the overhead).

To solve (1) in CGM method,

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

initial guess of $\bar{x}$ 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,

$$ R_k = b - A\bar{x}_k \quad \text{(V.90)} $$

The readers do not need to understand CGM to understand HHL. Interested readers may refer to the literature (e.g. [10]) for more details.

**ACKNOWLEDGMENT**

(Anika Zaman and Hector Jose Morrell contributed equally to this work.)

**REFERENCES**

[1] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in Proc. 35th Annu. Symp. Found. Comput. Sci., Nov. 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., vol. 11, p. e1481, Feb. 2021.

[3] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, “Quantum computing for finance: State-of-the-art and future prospects,” IEEE Trans. Quantum Eng., vol. 1, pp. 1–24, 2020, doi: 10.1109/TQE.2020.3030314.
[4] F. Arute, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019, doi: 10.1038/s41586-019-1666-5.

[5] L. S. Madsen, F. Lautenbach, M. F. Askarani, F. Rortais, T. Vincent, J. P. F. Bulmer, F. M. Miato, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, J. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, “Quantum computational advantage with a programmable photonic processor,” *Nature*, vol. 606, no. 7912, pp. 75–81, Jun. 2022, doi: 10.1038/s41586-022-04725-x.

[6] (2018). *National Strategic Overview for Quantum Information Science Report (National Science and Technology Council, 2018).* [Online]. Available: https://www.quantum.gov/wp-content/uploads/2020/10/2018_NSTC_National_Strategic_Overview_QIS.pdf

[7] N. D. Mermin, “Could Feynman have said this?” *Phys. Today*, vol. 57, no. 5, pp. 10–11, 2004, doi: 10.1063/1.1768652.

[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.

[9] Y. Cao, A. Daskin, S. Frankel, and S. Kais, “Quantum circuit design for solving linear systems of equations,” *Mol. Phys.*, vol. 110, nos. 15–16, pp. 1675–1680, Aug. 2012.

[10] R. Chandra, S. C. Eisenstat, and M. H. Schultz, “Conjugate gradient methods for partial differential equations,” in *Proc. AICAInt. Symp. Comput. Methods Partial Differ. Equ.*, Bethlehem, PA, USA, Jun. 1975, pp. 1–12.

[11] F. Gao, G. Wu, M. Yang, W. Cui, and F. Shuang, “A hybrid algorithm to solve linear systems of equations with limited qubit resources,” *Quantum Inf. Process.*, vol. 21, no. 3, p. 111, Mar. 2022, doi: 10.1007/s11128-021-03388-3.

[12] S. Dmitry, B. Brieler, and J. F. Scherrer, “The potential of quantum computing and machine learning to advance clinical research and change the practice of medicine,” *Missouri Med.*, vol. 115, no. 5, pp. 463–467, 2018.

[13] B. Duan, J. Yuan, C.-H. Yu, J. Huang, and C.-Y. Hsieh, “A survey on HHL algorithm: From theory to application in quantum machine learning,” *Phys. Lett. A*, vol. 384, no. 24, Aug. 2020, Art. no. 126595.

[14] 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, Jun. 2020.

[15] H. J. Morrell and H. Y. Wong, “Study of using quantum computer to solve Poisson equation in gate insulators,” in *Proc. Int. Conf. Simul. Semiconductor Processes Devices (SISPAD)*, Sep. 2021, pp. 69–72, doi: 10.1109/SISPAD54002.2021.9592604.

[16] P. Schleich. (2019). *How to Solve a Linear System of Equations Using a Quantum Computer.* [Online]. Available: http://www.acom.rwth-aachen.de/_media/3teaching/_00projects/schleich.pdf

[17] *HHL Example Using Qiskit.* Accessed: 2021. [Online]. Available: https://github.com/hyongwong/HHL_Example

[18] A. Vandenbrocque, “On quantum algorithms for solving linear systems of equations,” M.S. Semester Project I, Commun. Theory Lab., École Polytechnique Fédérale de Lausanne, Switzerland, 2019. [Online]. Available: https://adrienvdb.com/projects-and-reports/

**ANIKA ZAMAN** received the B.Sc. degree in electrical and electronics engineering from North South University, Bangladesh, in 2016. She is currently pursuing the Graduate degree in electrical engineering with San Jose State University.

**HECTOR JOSE MORRELL** received the Master of Science degree from San Jose State University, in 2021.

**HIU YUNG WONG** (Senior Member, IEEE) received the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 2006.

He is an Associate Professor and Silicon Valley AMDT Endowed Chair in Electrical Engineering, San Jose State University. From 2006 to 2009, he worked as a Technology Integration Engineer at Spansion. From 2009 to 2018, he was a TCAD Senior Staff Application Engineer at Synopsys. His research interests include the applications of machine learning in simulation and manufacturing, cryogenic electronics, quantum computing, reliability simulations, wide bandgap devices (such as GaN, SiC, Ga2O3, and diamond) simulations, novel semiconductor devices design and design technology co-optimization (DTCO). His works have produced one book, one book chapter, more than 100 papers, and ten issued patents.

Dr. Wong received the Curtis W. McGraw Research Award from ASEE Engineering Research Council in 2022, the NSF CAREER award and the Newnan Brothers Award for Faculty Excellence in 2021, and Synopsys Excellence Award in 2010. He is the author of the book, “Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps”. He is also one of the founding faculties of the Master of Science in Quantum Technology at San Jose State University.

*[Online]. Available: https://github.com/hyongwong/HHL_Example

**A. Zaman et al.: Step-by-Step HHL Algorithm**