Quantum HHL algorithm applied to electric circuit and line transmission wave

A. Mordetzki, E. Buksman, and A. Fonseca de Oliveira

Facultad de Ingeniería, Universidad ORT Uruguay, Montevideo, Uruguay.

Received 22 September 2022; accepted 21 May 2023

The HHL quantum algorithm [1] is a procedure that addresses the resolution of linear systems of equations (QLSP). Under certain conditions, the algorithm has a logarithmic order in the number of equations, better than the faster classical method. The algorithm manages to find a quantum state proportional to the solution vector, up to a normalization factor. The disadvantage is that to determine each of the coefficients of the solution vector, the algorithm's output quantum state must be determined with additional statistical methods, thus losing its exponential advantage. There are certain types of problems in which this disadvantage can be circumvented, the statistical treatment is unavoidable, but for certain cases such as electrical circuits, in which the main interest is to find only one of the currents, (for example the load current), we only need to measure one of the qubits of the solution state. In this article we solve the linear system associated with the currents of an electrical circuit with sinusoidal voltage, using the HHL algorithm, simulated in a Scilab numerical environment. An optimized example of an electrical line transmission wave in a real computer on the IBMQ platform, is also solved.

Keywords: Quantum algorithms; quantum linear solver; electric circuits.

DOI: https://doi.org/10.31349/RevMexFisE.20.020206

1. Introduction

Solving linear systems of equations is crucial in many areas of mathematics and engineering. The Quantum HHL algorithm (HHL), formulated by Aram Harrow, Avinatan Hassidim, and Seth Lloyd in 2009 [1] is the first quantum algorithm capable of solving this problem with an logarithmic order on N (at least for sparse matrix), where N is the number of equations, when the best classical method is of order N [2]. However, the HHL quantum algorithm has certain practical limitations. By using a classical method we may determine each of the elements of the solution vector, but with a quantum solution we can only obtain a quantum state that represents the solution vector. To determine each of the entries of the solution vector, it is necessary to apply statistical methods, thus losing its original exponential advantage. In order to improve these aspects, some variations of the algorithm have been proposed [3], but these variations lack the capacity of being able to solve the problem completely.

On the other hand, one could try to reduce the statistical inference, looking for problems in which only one of the unknowns of the system of equations is relevant. In this article we will study a typical engineering and physics problem that appears in any undergraduate university course, in which the system of equations comes from Kirchhoff's laws applied to an electrical circuit of linear elements such as resistors, capacitors and inductances with AC currents. The complexity of the circuit increases, alongside the number of meshes, resulting in higher computational power needed to solve the equations. Fortunately, in the case of electrical circuits, typically only finding the module and phase of the current trough the load impedance is relevant, which simplifies the statistics to be made afterwards by allowing to use some form of analysis only on one qubit.

This analysis is interesting for a quantum computing course, from the pedagogical point of view, since it solves a well-known problem as a simple electrical circuits, using a quantum algorithm.

In Sec. 2, after a brief introduction, the HHL algorithm with related algorithms are presented. In Secs. 3 and 4, the general solution for circuits and transmission line is presented. As a practical example, in Sec. 5 analyzes a concatenated mesh circuit of $N = 4$ with impedances with a single sinusoidal voltage source, using the HHL algorithm simulated in the Scilab numerical environment. As a second example, in Chapter 6, the electromagnetic wave problem of an not dissipated transmission line will be solved using the same algorithm on the Qiskit platform (IBMQ open platform) [4].

2. The HHL algorithm

2.1. Objective and drawbacks of the algorithm

Given the system of equations $A\left|x\right\rangle = \left|b\right\rangle$, where A is an hermitic matrix, (or can ve otherwise reconverted), and $\left|b\right\rangle$ a unitary vector state, one may be interested in determining the solution vector. Actually, the outcome of the algorithm is not the coefficients of the vector $\vec{x}$ but a quantum state analogous to the solution $\left|x\right\rangle = A^{-1}\left|b\right\rangle$, up to normalization. Specifically, the outcome of the algorithm will be a normalized quantum state of the form $\left|x\right\rangle/\|x\|$ in which the amplitudes shall represent the solutions of our system. This has the important setback, that in order to be able to determi ne the value of the amplitudes, one relays on statistically undesirable methods such as Quantum Tomography [5], thus losing the quantum speedup advantage.

However, there are certain problems in which one may not need to know all the amplitudes, but some expectation value of an operator acting on the solution vector, or other

problems in which only one of the variables is relevant. Implementing a classical algorithm for solving linear systems requires solving the system almost in its entirety to be able to find one of the variables, while in the quantum algorithm, when the state $|x\rangle/\|x\|$ is found, we may apply statistics after measuring only one qubit.

2.2. Previous algorithms

The HHL algorithm uses two quantum subroutines that help construct the result vector $|x\rangle$, the Quantum Fourier Transform algorithm (QFT) [5] and the Quantum Phase Estimation algorithm (QPE) [6]. We sumarize theese two algorithms in the next sections.

2.2.1. QFT Algorithm

The idea of a Quantum Fourier Transform (QFT) arises from the fact that classical Discrete Fourier Transform (DFT) [7], must be represented by a unitary operator, so it is straightforward to think of it as an implementable quantum circuit, acting on a quantum state. Therefore, by using universal quantum gates, this operator may be implemented on a real Quantum Computer.

Let $|X\rangle$ and $|Y\rangle$ be some certain Quantum State of the computational Basis the QFT is defined as:

$$U_{QFT}\,|X\rangle = \frac{1}{2^{t/2}}\sum_{y=0}^{2^t-1} e^{\frac{i2\pi xy}{2^t}}\,|Y\rangle\,, \tag{1}$$

where t is the number of qubits of the system, $|X\rangle$ and $|Y\rangle$ are computational basis states expressed in binary notation, x and y are the decimal number representation of such states.

After some algebraic treatment [5] the above Eq. (1) can be written as a separable form, acting on each qubit of the state $|X\rangle$:

$$\begin{aligned} U_{QFT}\,|X\rangle = \frac{1}{2^{t/2}}&\left(|0\rangle + e^{2\pi ix/2^1}\,|1\rangle\right) \\ &\otimes\left(|0\rangle + e^{2\pi ix/2^2}\,|1\rangle\right)... \\ &\otimes\left(|0\rangle + e^{2\pi ix/2^t}\,|1\rangle\right). \end{aligned} \tag{2}$$

In the current state of the art of Quantum computing, commonly called Noisy intermediate-scale quantum era (NISQ, [8]), applying this quantum algorithm on a real quantum computer requires finding a decomposition of the matrix associated to the transformation into a universal set of gates, whose combination of Kronecker products and compositions yields the desired matrix.

For the QFT algorithm, this decomposition in quantum gates is known [5], and requires a few basic gates: the Hadamard transform H and a controlled rotations CR_z around the z axis; R_z. Their matrix representations are shown in Eq. (3) and Eq. (4).

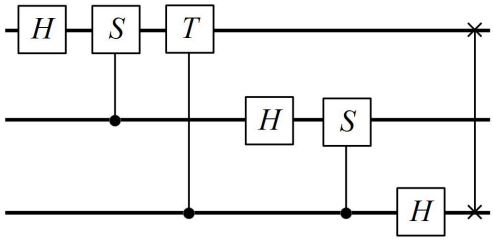

FIGURE 1. Three qubit QFT circuit.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{3}$$

$$CR_z = I\otimes I...\,|0\rangle\langle 0| + R_z(\phi)\otimes I...\,|1\rangle\langle 1|\,, \tag{4}$$

where $R_z(\phi)$ is

$$R_z(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}. \tag{5}$$

As an example, for a three qubit state, the QFT circuit with $R_z(\pi/2) = S$ and $R_z(\pi/4) = T$ is shown in Fig. 1.

2.2.2. QPE Algorithm

The Quantum Phase Estimation algorithm was devised to estimate the eigenvalues (or phases) of the corresponding eigenvectors of a unitary operator [5] .

For any unitary operator U, all the eigenvalues are of modulus 1, then for any $k \in 1,..,2^n$,

$$U\,|u_k\rangle = e^{2\pi i\theta_k}\,|u_k\rangle\,, \tag{6}$$

where $|u_k\rangle$ is a eigenvector of U. The QPE algorithm estimate the phase θ_k in t registers. Assuming that $0 \le \theta_k < 1$, and $2^t\theta_k \in \mathbb{N}$, then the algorithm found exactly the binary representation of θ_k, otherwise, it determines the value with high probability [see Eq. (12)].

The initial state $|\psi_0\rangle$ of the algorithm is

$$|\psi_0\rangle = |0\rangle^{\otimes t}\otimes|u_k\rangle\,. \tag{7}$$

The first step of the algorithm is to apply a set of Hadamard, $H^{\otimes t}$ to the t registers qubits, resulting in the state

$$|\psi_1\rangle = \frac{1}{2^{t/2}}\left(|0\rangle + |1\rangle\right)^{\otimes t}|u_k\rangle\,. \tag{8}$$

A set of controlled operators CU^{2^j} are then applied, where the controls attacks the register $j \in 0,...,t-1$. For example, for the third register CU^{2^2} is

$$\begin{aligned} CU^{2^2} = &|0\rangle\langle 0|\otimes I\otimes I\otimes I \\ &+ |1\rangle\langle 1|\otimes I\otimes I\otimes U^4. \end{aligned} \tag{9}$$

The complete representation of QPE algorithm is shown in the Fig. 2.

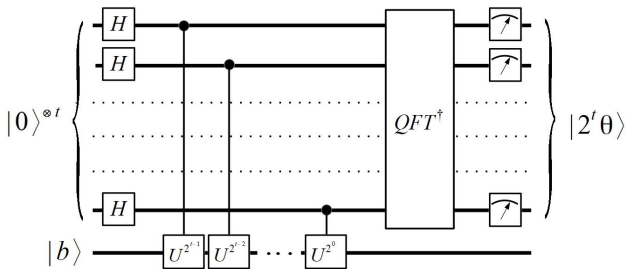

FIGURE 2. QPE algorithm.

The resulting state after some algebraic manipulations is

$$|\psi_3\rangle = \frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\theta_k}|1\rangle\right)\otimes ... \otimes\left(|0\rangle + e^{2\pi i 2^{0}\theta_k}|1\rangle\right)|u_k\rangle\,. \quad (10)$$

As we see from Eq. (2), if $2^t\theta_k \in \mathbb{N}$ and $0 < \theta_k < 1$ could be replaced by x getting exactly the Eq. (10), that can be rewritten as

$$|\psi_3\rangle = \left(U_{QFT}\left|2^t\theta_k\right\rangle\right)|u_k\rangle\,. \quad (11)$$

By applying an inverse transform $U^{\dagger}_{QFT}$ on the registers qubits we obtain the exact binary representation of the phase.

When $2^t\theta_k \neq \mathbb{N}$, several authors analyzed the error that this condition causes in the phases [5, 9]. In Ref. [5] the authors demonstrated that for any $m \in \mathbb{N}$ and $\epsilon > 0$, the phase value is obtained with m correct digits with probability at least $1-\epsilon$, chooses the number of register qubits as,

$$t \geq m + \log\left(2 + \frac{1}{\epsilon}\right). \quad (12)$$

2.3. HHL Algorithm

The previous algorithms are used as part of the HHL algorithm. To solve the system of equations $A|x\rangle = |b\rangle$, where A is a $2^n \times 2^n$ matrix with eigenvalues θ_k, and $|b\rangle$ a 2^n normalized vector. The initial state with $t+n+1$ qubits; t registers qubits, n qubits for the state $|b\rangle$, and one ancilla qubit, will evolved to the answer. The algorithm initialize with the state of the form

$$|\psi_0\rangle = |0\rangle|0\rangle^{\otimes t}|b\rangle\,, \quad (13)$$

where b can be decomposed on the basis eigenvectors of A as

$$|b\rangle = \sum_{k=1}^{2^n} b_k|u_k\rangle. \quad (14)$$

The t the qubits will be used to store the result of the phases.

The preparation of the state $|b\rangle$ is as computationally complex as the algorithm itself, for the sake of simplicity we assume the existence of an operator M, such that $M|0\rangle = |b\rangle$, is determined efficiently and without errors.

- *Step 1.* The QPE estimation (2.2.2.) is applied to the t registers qubits of initial state. After the first step the state becomes

$$|\psi_1\rangle = H^{\otimes t}|0\rangle^{\otimes t}|b\rangle = \frac{1}{2^{t/2}}(|0\rangle + |1\rangle)^{\otimes t}|b\rangle \quad (15)$$

- *Step 2.* To proceed, with the QPE estimation, the unitary $U = e^{i2\pi A}$ must be generated, this stage can be done with a procedure called Hamiltonian Simulation, which also requires some computational complexity discussed in Ref. [10]. As a consequence, the eigenvalues of A, $\{\theta_k\}$ have been turned into the phases of the operator U.

  After applying QPE estimation to the state $|\psi_1\rangle$ we get

$$|\psi_2\rangle = \sum_{k=1}^{2^k}\frac{1}{2^t/2}(|0\rangle + e^{2\pi i\theta_k 2^{t-1}}|1\rangle)\otimes ... \otimes(|0\rangle + e^{2\pi i\theta_k 2^1}|1\rangle)\otimes(|0\rangle + e^{2\pi i\theta_k 2^0}|1\rangle)b_k|u_k\rangle$$

  Then, an inverse of a Quantum Fourier Transform defined as Eq. (2) is applied to the resulting register in order to obtain an analogous state $|\theta_k\rangle$ where θ_k is the eigenvalue of the eigenstate $|u_k\rangle$ The resultant state is of the form:

$$|\psi_3\rangle = QFT^{\dagger}|\psi_2\rangle = \sum_{k=1}^{2^n}\left|\bar{\lambda}_k\right\rangle b_k|u_k\rangle\,, \quad (16)$$

  where $\bar{\lambda}_k$ is the binary approximation of the eigenvalue θ_k in the t registers. From now on, we will assume that the estimation is sufficient in its precision, and therefore we have the exact eigenvalues of the matrix A.

- *Step 3.* The idea behind the algorithm is to invert the eigenvalues to get:

$$A^{-1}|b\rangle = \sum_{k=1}^{2^n}\frac{1}{\bar{\lambda}_k}b_k|u_k\rangle. \quad (17)$$

  Once the eigenvalues are known, we proceed to apply a rotation around the y axis on the first auxiliary qubit

$$R_Y = e^{-iY\phi_k}, \quad (18)$$

  where $\phi_k = \arcsin\left(C/\bar{\lambda}_k\right)$ using the known estimation of eigenvalues on the registers t as controls, resulting in

$$|\psi_4\rangle = \sum_{k=1}^{2^n}|0\rangle\sqrt{1-\frac{C^2}{\bar{\lambda}_k^2}}\left|\bar{\lambda}_k\right\rangle b_k|u_k\rangle + |1\rangle\frac{C}{\bar{\lambda}_k}\left|\bar{\lambda}_k\right\rangle b_k|u_k\rangle\,, \quad (19)$$

where the constant C cancels out and ensures the rotation may be implemented. Taking $C = \min|\bar{\lambda}_k|$ is generally accepted.

Taking into account that if a measurement of the ancilla is made, and wait until the resulting state $|1\rangle$ is obtained, then we have already succeeded in inverting the corresponding eigenvalue corresponding to $|u_k\rangle$.

Now it only remains to apply the inverse of QPE onto the remaining state, resulting in

- *Step 4.*

$$|\psi_5\rangle = |1\rangle \otimes QPE^{\dagger} \sum_{k=1}^{2^n} \frac{1}{\bar{\lambda}_k} |\bar{\lambda}_k\rangle b_k |u_k\rangle$$
$$= |1\rangle \otimes |0\rangle^{\otimes t} \otimes \frac{|x\rangle}{\|x\|}, \quad (20)$$

and in principle we should obtain the solution state $|x\rangle$, up to a norm, in the qubits corresponding to $|b\rangle$.

2.3.1. *Order of HHL algorithm*

The HHL algorithm is capable of solving this problem with an order of [1]

$$O\left(\frac{\log(N)s^2k^2}{\epsilon}\right), \quad (21)$$

where k is the condition number of the matrix A, expressed as $k = \lambda_{\max}/\lambda_{\min}$ where $\lambda_{\max}$ y $\lambda_{\min}$ are the minimum and maximum eigenvalues of the matrix, s is the sparsity index of the matrix indicating the maximum number of non zero elements per file or column, and ϵ indicates the smallest measure of error in representing a real number.

Using the best classical method in terms of speed, requires an order of $O(Nsk\log(1/\epsilon))$ [11] thus, we notice an exponential upgrade in the number of equations N in using the HHL with regard to a classical method. However, we have a downgrade in k and s.

3. Solving *N* Mesh Circuit using HHL

Let any $N = 2^n$ (with n natural number) mesh electric circuit composed of linear elements such as resistors, inductance, and capacitance with sinusoidal voltage sources, can be solved using Kirchhoff's equations. If the number of meshes is not exactly 2^n, then the identity equations must be added.

For a stationary regime, the generalized Ohm's law is applied between two nodes l, m

$$I_{l,m} = \frac{V_{l,m}}{Z_{l,m}}, \quad (22)$$

where $V_{l,m}$ is the generalized complex voltage between the nodes, $Z_{l,m} = R_{l,m} + i(\omega L_{l,m} - [1/\omega C_{l,m}])$ is the impedance corresponding to that section, and $I_{l,m}$ is the generalized complex electric current.

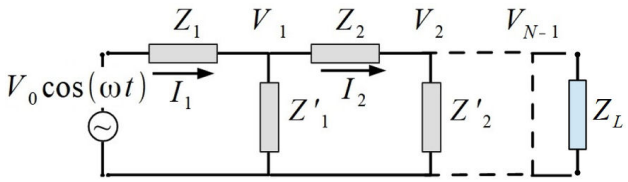

FIGURE 3. Example of N mesh electric circuit.

As a case of study, we will analyze a particular circuit, shown in Fig. 3.

Where extrapolating the voltage to complex values $V_0\cos(\omega t) \to V_0 e^{i\omega t}$, the corresponding N linear equations for the currents are:

$$V_0 - Z_1 I_1 - Z_1'(I_1 - I_2) = 0$$
$$-Z_2 I_2 - Z_2'(I_2 - I_3) + Z_1'(I_1 - I_2) = 0$$
$$\dots$$
$$-Z_L I_L + Z_{N-1}'(I_{N-1} - I_L) = 0. \quad (23)$$

For example, the matrix representation of the equations with $N = 2$ is

$$\begin{bmatrix} (Z_1 + Z_1') & -Z_1' \\ Z_1' & -(Z_1' + Z_L) \end{bmatrix} \begin{bmatrix} I_1 \\ I_L \end{bmatrix} = \begin{bmatrix} V_0 \\ 0 \end{bmatrix}.$$

4. Solving a Transmission Line Wave (TLW) using HHL

A transmission line is a one of the devices used to propagate electromagnetic waves. In general, it is a cable made with a good electrical conductor, long enough, compared to the wavelength size, in such way the phenomena of a wave nature to be relevant. These transmission lines are used to connect small wavelength radio transmitters to corresponding antennas, computer network connections, as well as in computer data buses of high frequency. An example of a typical electric transmission line is coaxial cable (see Fig. 4).

The transmission lines (TL) are a distributed parameter system on a line, but they can be analyzed as concatenated cells of infinitesimal length circuits, applying simple Kirch-

FIGURE 4. Typical electric line transmission used: a coaxial cable.

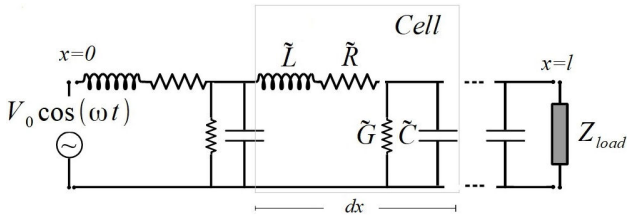

FIGURE 5. Detailed circuit model for Transmission Line cable.

hoff Laws. Usually, a given line transmission cable exhibits a certain electrical resistance, as well as an electrical inductance, capacitance and reluctance, these magnitudes are called parasitic as they were not put on purpose but are intrinsic properties of the cable.

Let us consider a transmission line cable of length l, as a set of infinitesimal cells of length dx. The infinitesimal resistance of the cell will be $\tilde{R} = R(dx/l)$, where R is the total parasitic resistance. In the same way the infinitesimal inductance will be $\tilde{L} = L(dx/l)$, where L is the total parasitic inductance, the infinitesimal capacitance $\tilde{C} = C(dx/l)$, where C is the total capacitance, and finally the infinitesimal reluctance $\tilde{G} = G(dx/l)$. An example of the circuit is shown in Fig. 5.

We study here an example of the propagation of a electromagnetic current wave on the TL cable without dissipation, means $R \to 0$ and $G \to \infty$ and a finite resistive load $Z_L = R'$ The wave equation for the currents at any point $0 \le x \le l$ and any times t is [12].

$$\frac{\partial^2 I}{\partial x^2} = \frac{1}{{v_w}^2}\frac{\partial^2 I}{\partial t^2}, \tag{24}$$

where $v_w = l/\sqrt{LC}$ is the propagation velocity of the electrical wave [12]. The expected wave solution, *i.e.* a current wave traveling to right and another one traveling to left for any particular frequency $\omega = 2\pi f$, and number wave $k = \omega/v_w$ will be

$$I(x,t) = ae^{i(\omega t - kx)} + be^{i(\omega t + kx)}, \tag{25}$$

where a and b are the complex amplitudes of the wave traveling to right and left respectively.

The frontiers condition for the left extreme of the lines ($x = 0$), using Kirchhoff's circuit law yield

$$i\omega V_0 e^{i\omega t} = -\frac{l}{C}\left(\frac{\partial I}{\partial x}\right)_{x=0}, \tag{26}$$

and for right extreme of the lines ($x = l$) Kirchhoff's circuit law yield

$$R'\left(\frac{\partial I}{\partial t}\right)_{x=l} = -\frac{l}{C}\left(\frac{\partial I}{\partial x}\right)_{x=l}. \tag{27}$$

Applying the conditions (26) and (27) to the Eq. (25) will get the lineal system of equations

$$\begin{aligned} a - b &= \frac{V_0}{Z_0}, \\ ae^{-ikl}(p-1) + be^{ikl}(p+1) &= 0, \end{aligned} \tag{28}$$

where $Z_0 = \sqrt{L/C}$ is called characteristic line impedance and $p = R'/Z_0$.

Then the matrix representation is:

$$\begin{bmatrix} 1 & -1 \\ e^{-ikl}(p-1) & e^{ikl}(p+1) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} V_0/Z_0 \\ 0 \end{bmatrix}. \tag{29}$$

5. HHL of electrical circuits in Scilab

To implement the HHL algorithm for 3-mesh electrical circuits or more, we need some elaborated quantum gates. Although it is possible to decompose them in basic gates (CNOT and generic one-qubit gates), the synthesis is still a complicated task for a general unitary matrix. The Qiskit platform is advancing in leaps and bounds, for example achieving efficient implementations of particular types of matrices. At the time of writing, (according to the tutorial [13]) the only truly efficient implementation it contains is the real Tridiagonal-Toeplitz matrix class. Therefore, because our matrix is of complex values, we give here a simulation of solving an electrical circuit equations with $N = 4$ variables, using HHL algorithm on Scilab platform.

As an example, the 3-mesh circuit Eq. (23) is analyzed. Because for quantum states the global phase is undefined, to obtain the global phase of the currents, one more identity equation must be added $V = V_0$ to fix the phase, then the equations becomes

$$\begin{aligned} V &= V_0, \\ V - (Z_1 + Z_1')I_1 + Z_1' I_2 &= 0, \\ Z_1' I_1 - (Z_2 + Z_2' + Z_1')I_2 + Z_2' I_L &= 0, \\ Z_2' I_2 - (Z_2' + Z_L)I_L &= 0. \end{aligned}$$

where all the impedance values Z_i, Z_i' were chosen with random phase and of order of magnitude 1 Ω.

The idea here is to calculate only the module and the phase of the load current I_L. To apply the HHL algorithm, the associated matrix must be hermitian. For the electrical circuit proposed, Eq. (30) the associated matrix A, of 4×4, is in general not hermitian, so the matrix has to be expanded to the form

$$A = \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix}, \tag{30}$$

where the 0 minds a quadratic matrix of zeros of size A, and the state $|b\rangle$ must be expanded accordingly to $|b, 0\rangle$. Then the new matrix A is finally 8×8.

The QPE phase estimation algorithm was designed to find the different eigenvalues of an hermitian matrix, satisfying $\lambda_i \in (0, 1)$. In general, the expansion of the matrix A Eq. (30) does not meet this condition, so this array must be modified together with the state b, in such a way that

$$\hat{A} = \frac{A}{\sqrt{\mathrm{Tr}(AA^\dagger)}}, \quad |\hat{b}\rangle = \frac{1}{\sqrt{\mathrm{Tr}(AA^\dagger)}}|b\rangle. \tag{31}$$

Note that, this mapping does not affect the currents since $|I\rangle = \hat{A}^{-1}|\hat{b}\rangle = A^{-1}|b\rangle$. For the simulation of the algorithm, $n = 3$ qubits were used for the expanded state $|\hat{b}\rangle$ and $t = 4, 5, 6, 7$ registers ancillaries qubits for the estimation of the eigenvalues.

First the QPE algorithm is applied to find the eigenvalues which will then determine the rotations Eq. (18) necessary for the inversion of the eigenvalues. One of the difficulties encountered in the application of QPE is that it can be demonstrated that the expansion Eq. (30) of the matrix leads to half of the eigenvalues being positive and the other half negative with, the same modulus. The latter causes that the phases obtained from the negative eigenvalues are shifted by 2π, resulting in the negative eigenvalues being also shifted by $-\lambda_i \rightarrow (-\lambda_i + 1)$. We assume that the eigenvalues are unknown and therefore we can't determine which ones were shifted. This becomes a problem in the next step of HHL algorithm, because we need the eigenvalues in order to apply the rotations Eqs. (18). One way of solving this problem is to first run QPE algorithm with a shifted matrix, *i.e.*:

$$\widetilde{A} = \frac{\hat{A} + k_d I}{\sqrt{\mathrm{Tr}\left(\hat{A}\hat{A}^{\dagger}\right)}}, \tag{32}$$

being $k_d = \max(|\lambda_i|) + 1/2^t$. In that way, the eigenvalues obtained are always positive and less than one. Of course, this means that we had to have a classical way of estimating the minimum eigenvalue of the matrix. After obtaining the eigenvalues, the inverse operation is performed, obtaining the corrected ones. Then, the corresponding rotations are applied and the resulting state is the expected currents. Besides that, if the resulting shifted eigenvalues overlap, the QPE algorithm gives wrong results, however that case is not addressed here.

At this stage, only the load current $I_L = |I_L|e^{i\phi}$ (module and phase) is relevant. To obtain it, we must add an ancillary qubit and apply an control operator M to the resulting state $|I\rangle$. In this case we choose the last equation of Eqs. (23) to be related to I_L, so the generalized M is a Toffoli operator

$$\begin{aligned} M = &|111\rangle\langle 111| \otimes X \\ &+ (|000\rangle\langle 000| + ... |110\rangle\langle 110|) \otimes I_d. \end{aligned} \tag{33}$$

where X is the Pauli matrix $X|0\rangle = |1\rangle$,$X|1\rangle = |0\rangle$ and I_d is the identity operator. After measuring the ancillary qubit in Z basis, we estimates the probabilities $P_Z(0)$ and $P_Z(1)$. The phase is also required, so we must measure also in X basis, obtaining $P_X(0), P_X(1)$, and the desired load current will be:

$$\begin{aligned} |I_L| &= P_Z(1), \\ \cos(\phi) &= \frac{P_X(0) - P_X(1)}{2P_Z(0)}. \end{aligned} \tag{34}$$

In order to determine the real load current I_L we must multiply first by the norm of the vector $|I\rangle$. This value cannot be obtained using the quantum algorithm so they have to be calculated utilizing some classical method.

5.1. Exact and approximate eigenvalues

A particular case arises when the eigenvalues of $\hat{A}$ take the exact values:

$$\lambda_i = \left\{0, \frac{1}{2^t}, \frac{2}{2^t}, ..., \frac{2^t - 1}{2^t}\right\}, \tag{35}$$

resulting in the procedure yielding an exact result for any given set of impedances, *i.e.* the gates do not induce error.

In contrast, when the eigenvalues do not fit the exact values of Eq. (35), the QPE approximates them to the closest binary value that depends on the number of registers t, hence the load current be affected due this approximation. In Table I we see how this approximation, produces errors in determining the load current $I_L = |I_L|e^{i\phi}$,

As an example, for source voltage fixed at $V_0 = 5V$, the resistance $R = 1\Omega$ as load impedance Z_L, and the other impedance set chosen as:

$$\begin{aligned} Z_1 &= 0.235 - 0.087i, \quad Z_2 = 0.561 + 0.600i, \\ Z_1' &= 0.302 + 0.781i, \quad Z_2' = 0.501 - 0.546i, \end{aligned} \tag{36}$$

the exact analytical load current is

$$I_L^{\text{clas.}} = 1.7476e^{i0.4867}.$$

The efficiency of the algorithm is measured with three parameters: the Fidelity between the result quantum state and the expected one, defined as $F(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|^2$ [5], that induces a distance between states

$$D_{HHL,\text{clas.}} = D(|\psi\rangle, |\phi\rangle) = \arccos(F(|\psi\rangle, |\phi\rangle)), \tag{37}$$

and the load current relative error in module ϵ(mod) and phase ϵ(phase).

As can be seen in Table I, although the distances between states $D(|\psi_{HHL}\rangle, |\psi_{\text{clas.}}\rangle)$ remain low as a function of t (on max order of 10^{-2}), as expected the error decreases as t increase, the load current modulus and phase have relatively high margin of errors for smaller t. This is due to the fact that the load is the last impedance in the circuit (3) and in general it is smaller, meaning that the largest error is precisely in the load current.

TABLE I. Resulting load current for set of impedance Eqs. (36), for different number of register t. The classical load current is $I_L^{\text{clas.}} = 1.7476e^{i0.4867}$.

t	I_L^{HHL}	$D_{HHL,\text{clas.}}$	ϵ(mod)	ϵ(phase)
4	$1.983e^{i0.47}$	4.2×10^{-2}	1×10^{-1}	3×10^{-2}
5	$2.291e^{i0.45}$	7.3×10^{-2}	2×10^{-1}	6×10^{-2}
6	$1.742e^{i0.49}$	2.4×10^{-3}	3×10^{-3}	6×10^{-4}
7	$1.749e^{i0.49}$	8.2×10^{-4}	2×10^{-5}	2×10^{-4}

6. HHL on Qiskit (IBM Q)

Although the Qiskit platform (IBMQ platform.) is continually advancing and complex unitary operator packages already exist, it is not yet possible to implement a general linear system of equations. Some intricate gates of many qubits, although already existing on the platform, are detailed here as a decomposition into CNOT and one-qubit gates.

The system of Eqs. (29) of electric transmission line has a 2×2 associated matrix A, that is not Hermitian in general. To get an Hermitian matrix, the original one must be expanded to 4×4 matrix. To simplify the problem, we take a particular case, where V_0, Z_0 are chosen free but $R_L = 2Z_0$ and $kl = \pi$, then $p = R_L/Z_0 = 2$, resulting in A a 2×2 Hermitian matrix equal to

$$A = \begin{bmatrix} 1. & -1. \\ -1. & -3. \end{bmatrix}. \tag{38}$$

The original matrix A has eigenvalues of modulus greater than one, so after divided by the factor $f = \sqrt{\mathrm{Tr}(AA^\dagger)}$, $\hat{A} = A/f$, $|\hat{b}\rangle = |b\rangle / f$, the new eigenvalues get in the range $[-1, 1]$. The implementation on IBM Q is now feasible with the basic gates defined in this platform.

First the QPE gate is applied to the initial 4-qubit state

$$|\psi_0\rangle = |a\rangle \otimes |t\rangle \otimes \left|\hat{b}\right\rangle = |0000\rangle\,, \tag{39}$$

where q_0 is the ancilla qubit, q_1 and q_2 are two registers $|t\rangle = |00\rangle$ and q_3 is the normalized vector $\left|\hat{b}\right\rangle = |0\rangle$. In the first part the QPE algorithm, the $CtrU$ gate have two Hadamard gates and the two controlled gates $C(U)$ and $C(U^2)$, where $U = e^{i2\pi\hat{A}}$ as shown in Fig. 6. The second part is the inverse of the QFT gate for two qubits.

After this application, one expected to get the approximation of the eigenvalues of $\hat{A}$, $\lambda_1 = -0.9342$ and $\lambda_2 = 0.3568$, in its binary form, in qubits q_1, q_2. For $t = 2$ registers, the approximation of the eigenvalues could only be the values: $\{0, 1/4, 2/4, 3/4\}$. Actually, the eigenvalues are in range $[-1, 1]$, but given the first eigenvalue λ_1 is negative (approximated -1),then the binary approximation is shifted by 1, getting the binary state $|00\rangle$ corresponding to the approximation of value 0. The second eigenvalue λ_2 has e binary approximation $|01\rangle$ corresponding $1/4$.

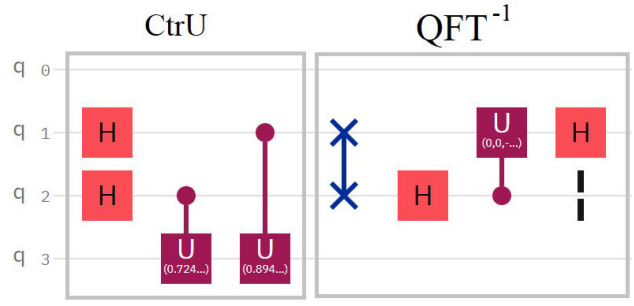

FIGURE 6. QPE algorithm with IBM Q quantum gates.

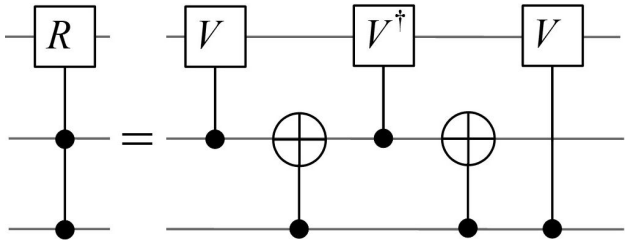

FIGURE 7. Generalized Toffoli gate for controlled ancilla rotations.

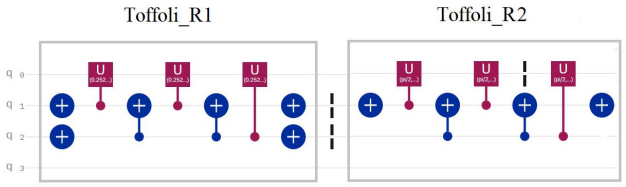

FIGURE 8. Rotations for inversion of eigenvalues on IBM Q.

The controlled ancilla rotations are generalized Toffoli gates: Toffoli_$R_{1,2}$, which could be decomposed as shown in Fig. 7 [5], where

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{40}$$

$$R_{1,2} = e^{-iY \arcsin(C/\lambda_{1,2})},\ C = 1/4 \tag{41}$$

$$V_{1,2} = \sqrt{R_{1,2}}. \tag{42}$$

The gates are shown in the IBMQ platform as in Fig. 8.

The desired coefficients a, b

$$|x\rangle = \begin{bmatrix} a \\ b \end{bmatrix} = \hat{A}^{-1} \left|\hat{b}\right\rangle, \tag{43}$$

calculated analytically are $a = \sqrt{9/10}, b = -\sqrt{1/10}$, given the probabilities of measuring in Z basis: $P_Z(0) = 0.9$ and $P_Z(1) = 0.1$. Due the estimation of eigenvalues by QPE, the expected coefficient (calculated by scilab) are not exact: $a = 0.9508, b = -0.3096$, being the expected probabilities being $P_Z(0) = 0.9041$ and $P_Z(1) = 0.0959$. The results in the IBMQ simulator for 32768 shots are $P(0) = 0.9023$ and $P(1) = 1 - P(0) = 0.0977$, within the range of statistical error. To get the phase of the coefficients we must measure also in X basis, obtaining finally the coefficients by IBM Q equal to: $a = 0.9499, b = -0.3125$. Finally, the same experiment with the same number of shots, but in a real IBM Q machine ibmq_quito, is running, and the coefficient obtained are $a = 0.9318, b = -0.3629$. The results are summarized in table Table II.

TABLE II. Coefficients a, b of electric transmission line wave.

$\|x\rangle$	Exact	ibmq_qasm_simulator	ibmq_quito
a	$\sqrt{9/10}$	0.9499	0.9318
b	$-\sqrt{1/10}$	-0.3125	-0.3629

7. Summary

In this article, the HHL quantum algorithm for solving linear system of equations is implemented for electrical circuits of sinusoidal current with N meshes and a current wave in a transmission line. In the case of electrical circuits with N meshes, only the load current is commonly relevant, but classically the system of equations must be solved almost in all its steps, resulting in methods of order$N$.

In the quantum case, the algorithm is logarithmic order in N, but has a drawback because the result is statistical in nature, that makes it to lose its advantage over the classical algorithm. In our case, we only want to find the load current, then in fact, the problem is limited to measurement and statistic of a single qubit. The study is carried out in a Scilab environment for $N = 4$ (which increases to $N = 8$ to convert the matrix into hermitian) obtaining the load current in module and phase. It can be seen in the Table II how increasing the registers qubits in the estimation of the eigenvalues in the QPE algorithm, improves the current result compared to standard results.

Finally, a case of transmission line is implemented on Qiskit platforms: $ibmq\_qasm\_simulator$ and on a IBM real quantum computer $ibmq\_quito$. The comparison between a classical linear solver, the simulated HHL algorithm and its implementation on real quantum hardware , yields similar results, thus, reaffirming the efficacy of the algorithm.

Acknowledgments

We acknowledge the use of IBM Quantum Services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

1. A. W. Harrow, A. Hassidim, S. Lloyd, Quantum Algorithm for Linear Systems of Equations. *Phys Rev Lett.* **103** (2009) 150502. https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

2. Y. Zheng *et al*., Solving Systems of Linear Equations with a Superconducting Quantum Processor. *Phys Rev Lett.* (2017) **118** 210504. https://link.aps.org/doi/10.1103/PhysRevLett.118.210504.

3. C. C. Chen, S. Y. Shiau, M. F. Wu, Y. R. Wu, Hybrid classical-quantum linear solver using Noisy Intermediate-Scale Quantum machines. *Scientific Reports.* **9** (2019) 16251. Available from: https://doi.org/10.1038/s41598-019-52275-6.

4. IBMQ. Quantum experience, https://quantum-computing.ibm.com; 2022. Available from: https://quantum-computing.ibm.com/2021.

5. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press; (2011).

6. A. Y. Kitaev Quantum measurements and the Abelian Stabilizer Problem. *Electron Colloquium Comput Complex.* (1996) 3.

7. C. Langtonand V. Levin The Intuitive Guide to Fourier Analysis and Spectral Estimation. Mountcastle Company, 1st edition; 2017.

8. Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, et al. Noisy intermediate-scale quantum algorithms. *Rev Mod Phys*. **94** (2022) 015004, https://link.aps.org/doi/10.1103/RevModPhys.94.015004.

9. J. M. Chappell, M. A. Lohe, L. von Smekal, A. Iqbal, D. Abbott, A Precise Error Bound for Quantum Phase Estimation. *PLOS ONE*. **6** (2011) 1-4. https://doi.org/10.1371/journal.pone.0019663.

10. D. W. Berry, G. Ahokas, R. Cleve, B. C. Sanders, Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Communications in Mathematical Physics*. **270** (2007) 359-71.

11. Saad Y. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics; (2003).

12. Alexander C, Sadiku M. Fundamentals of electric circuits. (McGraw-Hill Higher Education; 2007).

13. IBMQ. Qiskit Tutorial; 2022. Available from: https://qiskit.org/textbook/ch-applications/hhl_tutorial.html#4.-Qiskit-Implementation.