# A Quantum-Walk-Unitary HHL Matrix Equation Solver and Its Challenges in the NISQ Era

Xinbo Li*, Christopher Phillips$, Ian Jeffrey*, Vladimir Okhmatovski*
*Dept. of ECE, University of Manitoba, Canada
$Dept. of ECE, University of Waterloo, Canada
[1]lix34545@myumanitoba.ca, [2]c5phillips@uwaterloo.ca, [3]ian.jeffrey@umanitoba.ca
[4]vladimir.okhmatovski@umanitoba.ca

***Abstract* — The Harrow/Hassidim/Lloyd algorithm is a celebrated quantum matrix equation solver. Its Hamiltonian simulation involves a quantum walk process. A newly developed Quantum Walk Unitary HHL (QWU-HHL) leverages the spectral relationship between the quantum walk operator and the system matrix, and uses the quantum walk operator as its unitary in the phase estimation directly, allowing Hamiltonian simulation in the classical HHL to be removed, hence improving its efficiency. Despite the potential of being exponentially faster than classical matrix equation solvers, HHL is not feasible in the noisy intermediate-scale quantum era because its quantum circuit is too deep to preserve an accurate solution. In this work, we investigate the error behavior of QWU-HHL on a 7-qubit quantum system. Instead of executing the entire circuit, the behaviors of its sub-circuits which each contain only an initialization and a single gate are recorded. The error caused by isolated initialization and operation are analyzed.**

***Keywords* — HHL algorithm, quantum walk, NISQ hardware**

## I. Introduction

Matrix equations are ubiquitous in computational problems. The number of unknowns solvable by classical matrix algorithms has grown as computer hardware evolves. However, the polynomial cost associated with conventional solutions to matrix equations become prohibitive with large enough problem sizes. In computational electromagentics, a 3-D scattering problem ten wavelengths in size can result in millions of unknowns. Quantum computing provides a theoretically attractive alternative to perpetually seeking larger classical computing resources to solve larger problems. Specifically, the Harrow/Hassidim/Lloyd (HHL) algorithm [1] is a sparse quantum matrix equation solver that scales as $O(\log N)$, where $N$ is the number of linear equations. Compared with the optimal classical algorithm that scales as $O(N)$ (for sparse matrices with a constant number of unknowns per row), HHL provides exponential speedup.

HHL represents the solution to a matrix equation in terms of eigenvalues and eigenvectors of the system operator. Eigenvalues are computed using quantum phase estimation (QPE) by a Hamiltonian simulation which applies the unitary $e^{iAt}$ to $|b\rangle$ [2], where the implementation of the Hamiltonian simulation uses the quantum walk algorithm [3]. A recently-proposed improvement that we refer to herein as the quantum walk unitary HHL (QWU-HHL) [4] improves the efficiency of classical HHL by choosing the quantum walk operator as the unitary directly and removing the extra step of Hamiltonian simulation of $e^{iAt}$. While this improvement is theoretically promising, HHL and its variants are practically troubled as their implementation on noisy intermediate-scale quantum (NISQ) hardware has proven to be futile due to NISQ hardware quantum information loss, especially for deep circuits [5]. In an attempt to better understand the effects of quantum information loss during the QWU-HHL algorithm, in this work we subdivide the algorithm into a sequence of quantum gates after each of which the error can be assessed. This study effectively experimentally addresses the question of whether or not it is possible to work around deep circuit effects by subdividing and re-intializing the problem at each subdivision. This investigation is undertaken on a 7-qubit quantum processor.

## II. The Quantum Walk Unitary HHL Algorithm

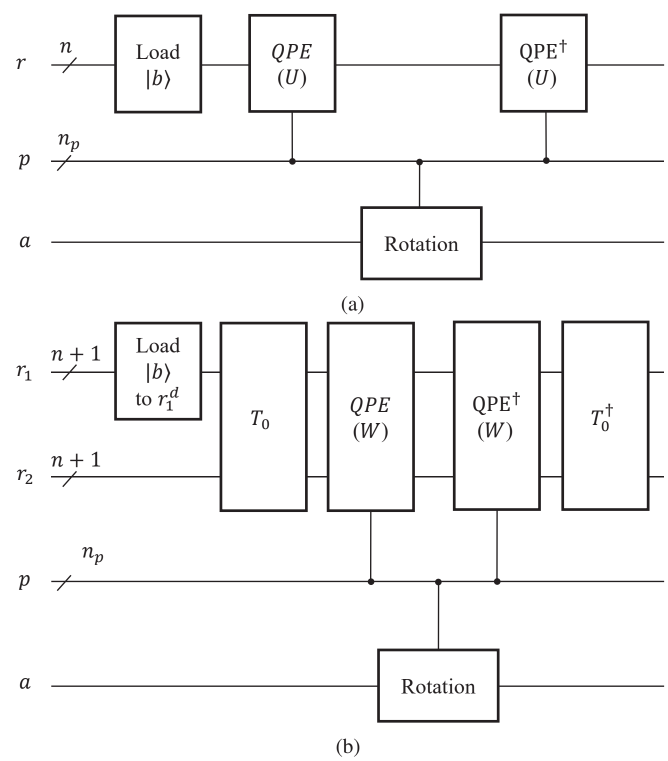


Fig. 1. HHL diagrams: (a) classical HHL; (b) QWU-HHL.

We seek to solve the matrix equation $A|x\rangle = |b\rangle$, where $A \in \mathbb{C}^{N\times N}$ is Hermitian and $N$ is a power of 2 such that $n = \log_2(N)$. If the matrix does not have these properties, an appropriately prepared matrix can be generated by straightforward augmentation of $A$ [4].

The quantum circuit diagram of QWU-HHL is shown in Fig.1b while the diagram of classical HHL is provided in Fig.1a for comparison. Comparing QWU-HHL to HHL we observe that QWU-HHL needs two $(n+1)$-qubit registers $r_1$ and $r_2$ to store the eigenvectors of the quantum walk operator $W$. In these registers, the last qubit is the ancilla qubit (denoted with superscript $a$) used to distinguish states in the first $n$ data qubits (denoted with superscript $d$). Hence, the states in registers $r_1$ and $r_2$ are represented as $|r_1\rangle = |r_1^d, r_1^a\rangle$, $|r_2\rangle = |r_2^d, r_2^a\rangle$. Besides $r_1$ and $r_2$, the remaining registers are the same as classical HHL, i.e., an $n_p$-qubit phase register and one qubit for the HHL ancilla.

Complete details of the QWU-HHL framework can be found in [4]. The difference between QWU-HHL and classical HHL are discussed in what follows, where QWU-HHL proceeds according to the following six steps:

1) Encode $|b\rangle$ into $|r_1^d\rangle$
2) Apply $T_0$ to $|r_1\rangle$ and $|r_2\rangle$
3) Perform the quantum phase estimation with $W$
4) Rotate the HHL ancilla controlled by $|p\rangle$
5) Invert the quantum phase estimation.
6) Invert $T_0$

Compared to the classical HHL [1] in Fig.1a, QWU-HHL has the additional steps of applying $T_0$ and its inverse (steps 2 and 6). In the classical HHL, these steps are unnecessary because the unitary $U = e^{iAt}$ and matrix $A$ have the same eigenvectors $|u_j\rangle, j = 0, 1, ...N-1$. Thus, in HHL, once $|b\rangle$ is encoded into register $r$ in the eigenbasis, $|r\rangle$ is in the superposition of the eigenstates of $U$:

$$|b\rangle = \sum_{j=0}^{N-1} \beta_j |u_j\rangle \tag{1}$$

In comparison, the QWU-HHL unitary is the quantum walk operator

$$W = iS(2TT^\dagger - I) \tag{2}$$

where the swap operator $S$ and operator $T$ are defined in [4]. Compared with $U$, the eigenvectors of $W$ have a more complicated relation with those of $A$. Therefore, to arrive at the superposition in the same form as in (1), we need to actively prepare the states in registers $r_1$ and $r_2$ as the eigenvectors of $W$. This is achieved by applying $T_0$ to these registers at step 2, where

$$T_0 = \sum_{j=0}^{N-1} \left( |j,0\rangle \langle j,0| \otimes B_j + |j,1\rangle \langle j,1| \otimes B'_j \right), \tag{3}$$

and the state preparation operators $B_j$ and $B'_j$ are defined as

$$B_j |0\rangle^{\otimes n} |0\rangle = |\phi_j, 0\rangle, \quad B'_j |0\rangle^{\otimes n} |0\rangle = |0\rangle^{\otimes n} |1\rangle. \tag{4}$$

Since we apply $T_0$ at the beginning of the algorithm, we invert it at the end (step 6).

Other steps of the algorithm are also present in the classical HHL. The difference is that the operations are tailored with the walk operator $W$ instead of the Hamiltonian evolution operator $U$. For example, the eigenvalues of $A$ (denoted by $\lambda_j$) are extracted from the eigenvalue phase of $W$ (denoted by $\theta_j$) stored in the phase register as

$$\lambda_j = X \sin(2\pi\theta_j) - d, \tag{5}$$

where $\theta_j \in [0, 1)$ is defined as $\mu_j = e^{2\pi\theta_j}$ for the eigenvalue $\mu_j$ of $W$, and $d$ is a pre-determined shift that arises from avoiding a branch cut in a square root [4]. The extracted eigenvalue will be used to calculate the angle of rotation applied to the HHL ancilla.

## III. Experiment and Results

At the time of writing, we have access to a 7-qubit quantum processor (IBM Jakarta). After assigning one qubit for each of $r_1^d, r_1^a, r_2^d, r_2^a$ and the HHL ancilla $a$, two qubits are left for the phase register $p$. According to the definitions in Section II, our quantum hardware limits us to consider the case $n = 1, n_q = 2, N = 2^n = 2$, i.e., a $2 \times 2$ matrix equation. For the purpose of demonstration, we choose a matrix equation that can be solved exactly by two phase qubits. That is, if we choose the eigenvalue phases of $W$ from the binary phases $0.00, 0.01, 0.10$, and $0.11$, they can be represented without error. Hence, we aim to design a $2 \times 2$ matrix equation that results in a $W$ with eigenvalues from this set. The system

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \tag{6}$$

with the choice $d = 3, X = 2$ produces the prepared matrix

$$A' = A + dI = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \tag{7}$$

which will be used to construct the quantum walk operator $W$. The eigenvalues of $A$ are $\lambda_0 = -1, \lambda_1 = -3$. When plugging the values of $\lambda, X$, and $d$ into (5), we find that the four eigenphases of $W$ are 0.00, 0.01 (repeated twice), and 0.10, indeed representable without error by two qubits. We use the normalized equal superposition of eigenvectors as the RHS vector, i.e., $|b\rangle = |1\rangle$.

The implementation of the algorithm amounts to applying the operators in Fig.1b in sequence, while each operator is constructed from a series of more fundamental quantum gates such as Hadamard gates, controlled gates, and rotation gates. For the specific $2 \times 2$ matrix equation, 120 fundamental gates are needed after problem-specific simplification. The entire HHL circuit of 120 gates is, after transpilation in terms of a given set of basis gates, too deep for the NISQ hardware, which motivates us to isolate each gate in action. This gate isolation allows us to observe the intermediate states in the entire quantum circuit, facilitating the study of error occurring at each gate. As a result of the gate isolation, the entire quantum circuit is divided into 120 sub-circuits, where in each sub-circuit only one gate is applied, as shown in Fig. 2. The state vector at the end of each sub-circuit is recorded using the

Qiskit state vector simulator to initialize the next sub-circuit. On the Jakarta backend, each sub-circuit is measured both right after the initialization and following the application of the single gate. The measurement results are compared with their simulated counterparts to benchmark the error of the initialization and the entire sub-circuit.

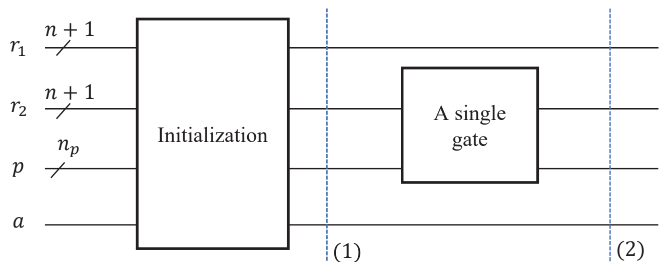


Fig. 2. A sample sub-circuit. Each sub-circuit contains two operations: the initialization that operates on all registers to set the state vector to a target state vector (the final state vector of the last sub-circuit) and a single fundamental gate applied to the work register. The sub-circuit is sliced at (1) and (2), the qubit states at these check points are recorded from both a simulator and the real quantum backend: IBM Jakarta.

A quantum circuit is transpiled before it is executed on a backend. The IBM Jakarta has six basis gates: $CX, I, R_z, \sqrt{X}, X$, and the if-else gate, where the first five are picked to transpile the circuit. The basis gate count of each transpiled sub-circuit is given in Fig. 3. We observe that

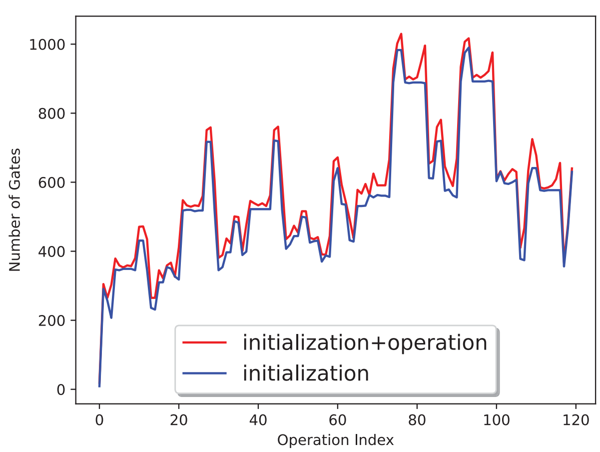


Fig. 3. The basis gate count of sub-circuits when transpiled with five basis gates on IBM Jakarta.

the initialization is an expensive process–the gate count due to initialization dominates the total gate count in each sub-circuit. As more gates means more operation time, causing a larger possibility of decoherence [6] (the major source of quantum information loss), the fact that the initialization requires more basis gates hints that the initialization is the main error source in a sub-circuit's initialization and single gate operation. Fig. 4 confirms this. In fact, an entire sub-circuit sometimes provides more accurate results than the initialization part of that sub-circuit. The last sub-circuit yields a final probability

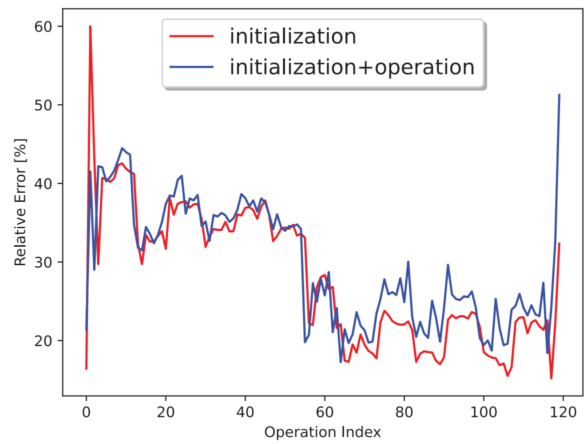


Fig. 4. The relative error of the probability vector generated from IBM Jakarta when compared with its simulated counterpart.

vector that is less than 50% accurate. This error is too high for any practical implementation of the algorithm in the quantum hardware, especially considering how small the problem is. In contrast, the simulated solution provides an average relative error of $1.125 \times 10^{-10}$.

## IV. Conclusion

It has been previously reported that QPE (an integral part of HHL) is not feasible on NISQ hardware [5]. Our experimental findings for QWU-QPE further confirm this, indicating that the initializing error required when decomposing the circuit into sub-circuits is catastrophic. These conclusions may change in the future, should a reliable quantum initialization routine be found. We observe that the re-initialization and division routine is not effective with Qiskit's default initialization functionality. The issue is that initializing the qubits into target states requires more gates than the gate count threshold that limits the accuracy. Other initialization techniques and quantum-classic hybrid systems will be investigated in future work.

## References


[1] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, oct 2009. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.103.150502

[2] A. M. Childs, "On the relationship between continuous- and discrete-time quantum walk," *Communications in Mathematical Physics*, vol. 294, no. 2, pp. 581–603, oct 2009. [Online]. Available: https://link.springer.com/article/10.1007/s00220-009-0930-1

[3] D. Berry and A. Childs, "Black-box hamiltonian simulation and unitary implementation," *Quantum Information and Computation*, vol. 12, no. 1-2, pp. 29–62, Jan. 2012.

[4] C. D. Phillips and V. I. Okhmatovski, "A quantum computer amenable sparse matrix equation solver," 2021. [Online]. Available: https://arxiv.org/abs/2112.02600

[5] R. Yalovetzky, P. Minssen, D. Herman, and M. Pistoia, "Nisq-hhl: Portfolio optimization for near-term quantum hardware," *arXiv preprint arXiv:2110.15958*, 2021. [Online]. Available: https://arxiv.org/abs/2110.15958

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.