Solution of 2x2 Matrix Equation on NISQ Quantum Computer with Quantum-Walk-HHL Algorithm

Christopher D. Phillips*, Helia Akbari†, and Vladimir I. Okhmatovski‡, *Member, IEEE*

*Brandon University, Brandon, MB R7A 6A9 Canada (email: phillipsch@brandonu.ca)

†Amirkabir University of Technology, Tehran, Iran (email: heliaakbari.edu@gmail.com)

‡University of Manitoba, Winnipeg, MB R3T 2N2 Canada (email: vladimir.okhmatovski@umanitoba.ca)

***Abstract*—Case study of solving a 2x2 matrix equation on a noisy intermediate-scale quantum (NISQ) quantum computer is presented. The structure of the matrix equation is customized for recently proposed quantum-walk Harrow/Hassidim/Lloyd (QW-HHL) algorithm in such a way that it minimizes the depth of the pertinent quantum circuit. Simplified operators in the QW-HHL algorithm are described for the particular 2x2 matrix equation to facilitate understanding of the method. Without uncomputation stage of the HHL algorithm, the 7 cubit circuit consists of 63 logical steps and features only 1,600 basic gates in the transpiled circuit. Such shallow depth of the circuit allows us to preserve quantum state containing the solution with fidelity of about 70% in the execution on IBM Fez quantum computer.**

***Index Terms*—Harrow/Hassidim/Lloyd (HHL), matrix methods, numerical analysis, quantum computing, simulation.**

I. Introduction

Quantum computation offers a promising alternative to classical computing methods in many areas of numerical science, with algorithms that make use of the unique ways in which quantum computers store and manipulate data often achieving dramatic improvements in performance over their classical counterparts. We develop a generalization of the Harrow/Hassidim/Lloyd algorithm by providing an alternative unitary for eigenphase estimation. This unitary, which we have adopted from research in the area of quantum walks, has the advantage of being well defined for any arbitrary matrix equation, thereby allowing the solution procedure to be directly implemented on quantum hardware for any well-conditioned system. The procedure allows for the inverse of a matrix to be applied with $\mathcal{O}\left(N_{nz}\log\left(N\right)\right)$ complexity, where N is the number of unknowns, and N_{nz} is the total number of nonzero elements in the system matrix. This efficiency is independent of the structure of the input matrix. Hence, this procedure represents an important step towards the achievement of quantum advantage over classical algorithms for practical problems in numerical science.

In this work, we study a specialized 2x2 matrix equation which is solvable on an actual quantum computer. The matrix equation and the QW-HHL algorithm are trivialized simplify its pertinent operators and minimize the depth of the quantum circuit. The depth of 63 logical steps in the pertinent 7 cubit quantum circuit implementation of the QW-HLL algorithms (without its uncomputation stage) allows us to maintain a near 70% fidelity in the quantum state carrying the solution. Experimentation demonstration of the circuit performance is conducted on 7 cubits of the IBM Fez quantum computer.

While the demonstrated problem size is necessarily small, the results provide valuable insight into how matrix-equation solvers based on HHL-type algorithms may be adapted for practical use on NISQ hardware. The study confirms that careful co-design of the numerical problem, the quantum algorithm, and the circuit implementation is essential for experimental viability. Although current hardware limitations restrict such demonstrations to toy-scale systems, the methodology developed here can be systematically extended to larger problems as quantum devices improve in qubit count, connectivity, and gate fidelity. Overall, this work serves as a concrete benchmark for implementing quantum linear-system solvers on real quantum hardware and offers a pathway for future development of scalable quantum matrix solvers. As quantum processors continue to mature, the QW-HHL framework and the circuit-simplification strategy demonstrated here are expected to play an important role in advancing quantum advantage for practical problems in computational science at large and computational electromagnetics (CEM) in particular.

II. Preliminaries of HHL Matrix Equation Solver

We begin with a matrix equation $A\left|x\right\rangle = \left|b\right\rangle$, where A is an $N\times N$ matrix, and $\left|x\right\rangle$ and $\left|b\right\rangle$ are N-dimensional vectors. If a quantum system is initialized to the state $\left|b\right\rangle$, then its state will be a superposition of the eigenvectors of A:

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

where $\left|u_j\right\rangle$ is an eigenvector of A, and β_j is the component of $\left|b\right\rangle$ along $\left|u_j\right\rangle$. We assume that the supplied matrix equation has been properly prepared:

1) $\left|b\right\rangle$ must be normalized.
2) A must be Hermitian.
3) N must be a power of two.
4) The eigenvalues of A must lie on the interval $[-X, X]$, where $X \geq N\max_{j,k}\left(\left|A_{jk}\right|\right)$.
5) No negative real numbers may appear on the diagonal of A.

To address point 5, a shifted matrix can be used:

$$\left(A + dI\right)\left|x\right\rangle = \left|b\right\rangle, \tag{2}$$

where d is an upper bound on the magnitude of the offending values on the diagonal of A. Methods to satisfy all the constraints for any arbitrary matrix is explained in reference [1], which is our Complete paper.

The solution vector $|x\rangle$ can then be obtained by applying the inverse of each eigenvalue λ_j of A to its corresponding eigenvector in the superposition $|b\rangle$:

$$\sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle = A^{-1} |b\rangle = |x\rangle \,. \tag{3}$$

This application of inverse eigenvalues, conditioned on the presence of the corresponding eigenvectors, is the basic function of HHL [2]. By using quantum phase estimation (QPE) [3], it is possible to extract and apply these eigenvalue factors in an efficient manner. However, this phase estimation requires the implementation of a problem-dependent unitary in terms of basic quantum gates. The process of decomposing this unitary into a sequence of basic quantum gates is nontrivial.

In this work we consider a particular method, inspired by research into quantum walks, which produces a unitary with a known decomposition into basic gates. Using the Qiskit SDK [4], we have developed a program [5] which closely follows the herein described procedure.

This procedure consists of four main stages: the initial application of T_0, quantum phase estimation, HHL rotation, and uncomputation.

III. A Problem Suitable for NISQ Quantum Systems

Given the extraordinary limitations on circuit size imposed by compounding gate errors, we must consider only the smallest of possible problems.

A. Problem Formulation

We choose a problem of two unknowns and a phase register of two qubits. The eigenphases of the system matrix must correspond exactly to the realizable values of a minimal phase register, to minimize the complexity of the QPE procedure. Therefore, the eigenvalues of the system matrix must satisfy

$$\lambda_j = X \sin(2\pi\varphi_j) - d = \begin{cases} -d, & \varphi_j = 0, \frac{1}{2} \\ X - d, & \varphi_j = \frac{1}{4} \\ -X - d, & \varphi_j = \frac{3}{4}. \end{cases} \tag{4}$$

The system

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}, \quad d = 3, \quad X = 2, \tag{5}$$

satisfies the constraints of our procedure, and has eigenvalues

$$\lambda_0 = -3 = -d, \quad \lambda_1 = -1 = X - d. \tag{6}$$

For a right-hand side vector, we use an equal superposition of both eigenvectors of A:

$$|b\rangle = \frac{1}{2}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{7}$$

For this problem of known form, some simplification is possible in the various operators of the solution procedure.

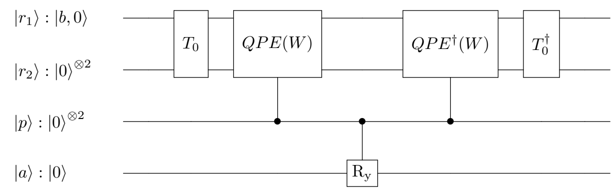

Fig. 1: Overview of the procedure for an example for 2 unknowns and size of the phase register equal to 2.

This allows the size of the final circuit to be significantly reduced. The system matrix defining the walk operator is

$$A + dI = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{8}$$

In Fig. 1, the procedure for this trivial example is illustrated as a high-level circuit. In the following sections, we will break down the implementation of each of these components.

B. Initialization of $|b\rangle$

The first step in the procedure is to initialize $|r_1\rangle$ to state $|b, 0\rangle$. Vector $|b\rangle$ in eq. 7 is equal to state $|0\rangle$. Given that qubits in quantum circuits are initially at state $|0\rangle$, there is no need for any additional gates to prepare $|b\rangle$.

C. T_0 Simplified

We first consider a state preparation operator B_j which prepares $|\phi_j^a\rangle$ from the $|0\rangle$ state, where

$$|\phi_j^a\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \left[\sqrt{\frac{N}{X} A_{jk}^*} |0\rangle + \sqrt{1 - \frac{N}{X} |A_{jk}|} |1\rangle \right]. \tag{9}$$

Given that in our simple example, $A_{jk} = 1$ for $j, k = \{0, 1\}$, and $N = X = 2$, $|\phi_j^a\rangle$ can be reduced to:

$$|\phi_0^a\rangle = |\phi_1^a\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) |0\rangle = H |0\rangle |0\rangle \tag{10}$$

and B_j can be implemented as a simple Hadamard gate.

T_0 can be effectively applied by conditional application of the state preparation operators. T_0 applies B_j to $|r_2\rangle$ with the control condition that $|r_1\rangle$ is in the state $|j, 0\rangle$.

$$T_0 = \sum_{j=0}^{N-1} \left(|j,0\rangle \langle j,0| \otimes B_j + |j,1\rangle \langle j,1| \otimes B_j' \right). \tag{11}$$

Since the initial state is supplied with ancilla state $|0\rangle$, it is possible to ignore the applications of B_j'. Therefore, T_0 can be simplified to:

$$T_0 = \sum_{j=0}^{N-1} \left(|j\rangle \langle j| \otimes B_j \right). \tag{12}$$

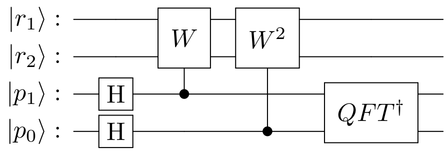

Fig. 2: Diagram of 2-qubit phase estimation using the walk operator.

Given that B_0 and B_1 are both equal to H, there is no need for conditional application. Therefore,

$$T_0 = (|0\rangle\langle 0| + |1\rangle\langle 1|) \otimes B_0 = I \otimes B_0 = B_0 = H \quad (13)$$

The T_0 operator can be implemented by simply applying a Hadamard gate to $|r_2\rangle$.

D. QPE Simplified

The walk operator that acts as the Hamiltonian for our QPE, which is illustrated in Fig. 2, is defined as:

$$W = iS\left(2TT^\dagger - I\right). \quad (14)$$

The S operator is the swap operator, which swaps the content of $|r_1\rangle$ and $|r_2\rangle$:

$$S = \sum_{j^a=0}^{2N-1}\sum_{k^a=0}^{2N-1} |k^a\rangle|j^a\rangle\langle j^a|\langle k^a|. \quad (15)$$

In our small example, this operation will have the effect

$$|r_1^0, r_1^a\rangle|r_2^0, r_2^a\rangle \xmapsto{S} |r_2^0, r_2^a\rangle|r_1^0, r_1^a\rangle \quad (16)$$

The $2TT^\dagger - I$ operator can then be implemented as

$$\begin{aligned} &2TT^\dagger - I \\ &= \sum_{j=0}^{N-1}\Big[|j,0\rangle\langle j,0| \otimes \left(2|\phi_j^a\rangle\langle\phi_j^a| - I\right) \\ &\qquad + |j,1\rangle\langle j,1| \otimes \left(2|\zeta_j^a\rangle\langle\zeta_j^a| - I\right)\Big]. \\ &= \sum_{j=0}^{N-1}\Big[|j,0\rangle\langle j,0| \otimes \left(B_j\left(2|0\rangle\langle 0| - I\right)B_j^\dagger\right) \\ &\qquad + |j,1\rangle\langle j,1| \otimes \left(B'_j\left(2|0\rangle\langle 0| - I\right)\left(B'_j\right)^\dagger\right)\Big]. \quad (17) \end{aligned}$$

$|\zeta_j^a\rangle$ correspond to failure states. B'_j prepares $|\zeta_j^a\rangle$ from $|0\rangle$. The B'_j operators can be implemented by simply switching the ancilla state of the operand register to $|1\rangle$. The circuit for the simplified walk operator is illustrated in Fig. 3.

E. the HHL Rotation

In this step, the ancilla register is used to impose the inverse eigenvalue factors on the system. We use the rotation operator

$$R_y\left(2\arccos\left(\frac{C}{\tilde{\lambda}_k}\right)\right)|0\rangle = \frac{C}{\tilde{\lambda}_k}|0\rangle + \sqrt{1 - \frac{C^2}{\tilde{\lambda}_k^2}}|1\rangle, \quad (18)$$

where C is a constant chosen to ensure that all the arguments in the arccos function obey its domain restrictions.

In our simple example, $C = 0.99$ and according to 4, the values of $\hat{\lambda}_k$ for $k = 0,1,2,3$ are, respectively, equal to $-3, -1, -3, and - 5$. The resulting circuit for R_y in Fig. 1 is illustrated in Fig. 4.

Finally, by applying the inverse of QPE and T_0, the uncomputation is implemented. The final result is equal to

$$C|x,0\rangle|0\rangle^{\otimes(n+1)}|0\rangle^{\otimes n_p}|0\rangle + |\text{failure}\rangle. \quad (19)$$

The final state after the implementation of the circuit for the trivial example is

$$0.99\left(-\frac{2}{3}|0,0\rangle - \frac{1}{3}|1,0\rangle\right)|00\rangle|00\rangle + |\text{failure}\rangle. \quad (20)$$

IV. Results

The simplification of the B_j operators carries through the T_0 and W operators. Without this reduction, the full program produces a circuit consisting of 20,616 basic gates after transpilation. After simplification, the final circuit requires only 3,111 gates. This is a notable reduction, but the circuit is still much too large considering the errors incurred by each gate application. These figures are taken from the final transpiled form of the circuit. The circuit was run on the Fez system operated by IBM Quantum [6]. 4096 individual trials, or "shots", were used for each data point. The results of these calculations are shown in Fig. 5.

Despite the prohibitive circuit size, we can still study the propagation of error throughout the procedure by a process of gradually adding minimal components of the full solution circuit, and tracking the overall error after each component's introduction. In total, the circuit was divided into 125 minimal-step components. An exact program detailing the individual operations can be found in [5].

Additionally, we have analyzed the quantum computer's output in the stage after the HHL rotation. The result of which can be observed in Fig.6. In this diagram, the states with HHL ancilla in state $|1\rangle$ have been ommitted as they are the failure state and will not contribute to the final correct answer. On top of the visible correspondence of output distribution (with 8192 shots) with the ideal simulation's probability distribuition, the color-coded sections inidicate different states of the phase register. High distribution can be seen on the green area where phase register is in the state $|01\rangle$ which corresponds to the eigenphase related to the eigenvalue λ_1.

In developing suitable tests for the practical performance of this procedure, it must be considered that errors on quantum systems can propagate rapidly. Therefore, when considering a problem suitable for fully quantum analysis, we are at present limited to very short, small calculations. It is possible to construct a more satisfying test using classical simulation. By using a classical computer to simulate a quantum system, gate errors can be eliminated, albeit at the cost of inefficient classical simulation.

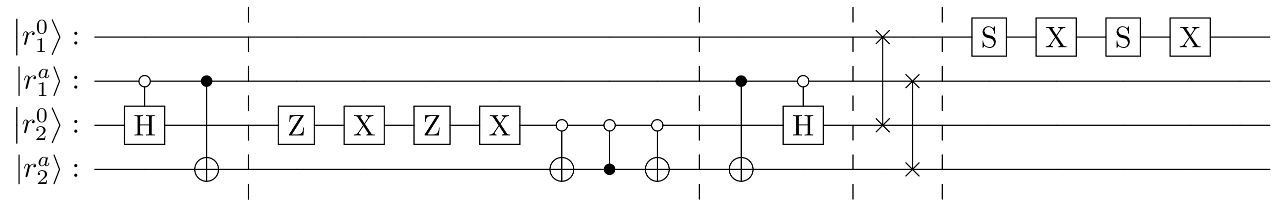

Fig. 3: Diagram of the walk operator $W = iS\left(2TT^\dagger - I\right)$ for the trivial example. The sections are as follows: I) controlled H for $B_j^\dagger$ and controlled X for $(B_j')^\dagger$, II) reflection about $|0\rangle$, III) controlled H for B_j and controlled X for B_j', IV) the SWAP operation, V) global phase i.

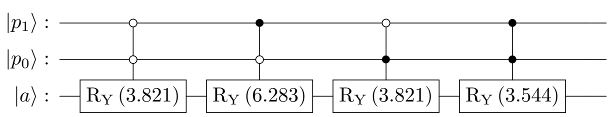

Fig. 4: Diagram of HHL rotation for the trivial example.

-

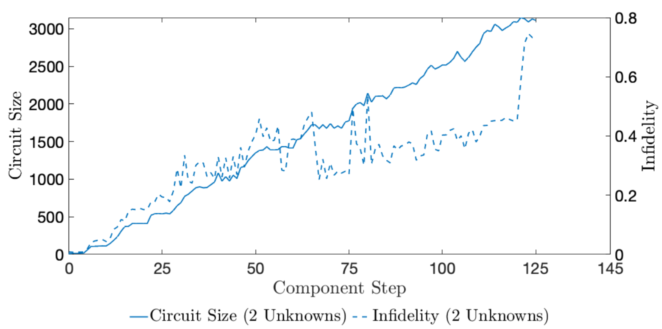

Fig. 5: Transpiled circuit size and infidelity of quantum state after execution.

V. Conclusion

In this work, we presented a detailed case study of solving a 2×2 matrix equation on a noisy intermediate-scale quantum (NISQ) computer using a quantum-walk–based generalization of the Harrow–Hassidim–Lloyd (QW-HHL) algorithm. By carefully tailoring both the problem formulation and the algorithmic operators, we demonstrated that the depth of the resulting quantum circuit can be significantly reduced without altering the fundamental structure of the method. For the considered example, this led to a seven-qubit circuit with only 63 logical steps and approximately 1,600 basic gates after transpilation when the uncomputation stage was omitted, making experimental execution on existing quantum hardware feasible. The simplified construction highlights how quantum-walk–inspired phase estimation can provide a practical alternative to conventional HHL implementations on NISQ devices. In particular, the explicit simplification of the state-preparation and walk operators enabled us to drastically reduce circuit complexity compared to a direct implementation, which would otherwise be prohibitively deep. Experimental execution on the IBM Fez quantum computer yielded a solution state with fidelity on the order of 70%, demonstrating that meaningful information can still be extracted from shallow quantum circuits despite accumulated gate errors and decoherence. The incremental error analysis further illustrates how circuit depth directly impacts solution fidelity and underscores the importance of operator-level simplifications for near-term quantum algorithms.

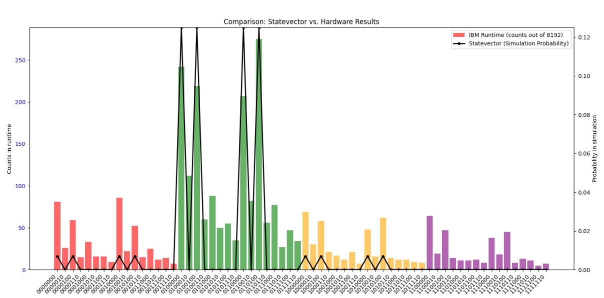

Fig. 6: Outcome of run on Fez before uncomputation.

Acknowledgment

We acknowledge the use of IBM Quantum services for this work.

References

[1] C. D. Phillips and V. I. Okhmatovski, “A Quantum Computer Amenable Sparse Matrix Equation Solver,” arXiv:2112.02600 [quant-ph], 2022. [Online]. Available: https://arxiv.org/abs/2112.02600

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150502, Oct. 2009, DOI: 10.1103/PhysRevLett.103.150502.

[3] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, “Quantum algorithms revisited,” *Proc. R. Soc. Lond. A.*, vol. 454, no. 1969, pp. 339–354, Jan. 1998, DOI: 10.1098/rspa.1998.0164.

[4] M. S. Anis *et al.*, “Qiskit: An Open-source Framework for Quantum Computing,” 2021, DOI: 10.5281/zenodo.2573505.

[5] C. D. Phillips, H. Akbari, and V. I. Okhmatovski, “QW-HHL,” 2026. [GitHub repository]. Available: https://github.com/heliaakbari/QW-HHL.

[6] IBM Quantum. https://quantum-computing.ibm.com/, 2026.