2022 IEEE Latin America Electron Devices Conference (LAEDC), Puebla, México, July 4-6, 2022

Study of Error Propagation and Generation in Harrow-Hassidim-Lloyd (HHL) Quantum Algorithm

Anika Zaman
Electrical Engineering
San Jose State University
San Jose, CA, USA
anika.zaman@sjsu.edu

Hiu Yung Wong*
Electrical Engineering
San Jose State University
San Jose, CA, USA
hiuyung.wong@sjsu.edu

***Abstract*— In this paper, we study the error propagation and generation in the Harrow-Hassidim-Lloyd (HHL) quantum algorithm runs on IBM-Q hardware with the help of a MATLAB simulator. HHL is a quantum algorithm that can provide exponential speedup over the fastest classical algorithm (conjugate gradient method) in solving systems of linear equations (SLE). However, without error correction, it cannot give correct results even in a 2-variable system due to its complexity. In this study, an HHL quantum circuit for a 2-variable SLE is implemented in IBM-Q and the error is extracted after each stage of the circuit and compared to a MATLAB simulator. We identified three major sources of errors, namely single-qubit flipping, gate infidelity, and error propagation. We also found that at the ancillary bit rotation stage, the error becomes large but the encoded solution still has high fidelity. However, the solution is mostly lost after the inverse quantum phase estimation which is necessary to extract the solution efficiently. Therefore, it is suggested that error correction resources, if limited, should be added to the second half of the circuit.**

Keywords—Error, HHL, Quantum Algorithm, Quantum Computing, Qubit Flipping

I. Introduction

Quantum computing is expected to solve many computationally challenging problems in classical computing such as factorization [1], quantum simulation [2], and optimization [3]. Quantum supremacy also has been demonstrated recently [4]. Among the important quantum algorithms, the Harrow-Hassidim-Lloyd (HHL) quantum algorithm promises to provide an exponential speedup in solving systems of linear equations (SLE) [5][6] and has a time complexity of $O(\log N)$, while the fastest classical algorithm, namely the conjugate gradient method, has a complexity of $O(N)$ [7]. An SLE can be represented as $\boldsymbol{A}\vec{x}=\vec{b}$, in which vector $\vec{x}$ is solved for a given symmetric matrix, $\boldsymbol{A}$, and a vector, $\vec{b}$. HHL algorithm can be used in machine learning [8], quantum system modeling [2][9], and solving poisson equations [10][11].

However, it is well-known that quantum computers are very susceptible to noise and even a simple SWAP gate implemented by 3 CNOT gates has more than 10% of error in a superconducting qubit computer [12]. Although it is expected that error correction is possible by implementing multiple physical qubits as one fault-tolerant logical qubit in the future [13], it is desirable to understand the source of the error in an algorithm and the parts of the algorithm that are the most affected by the error. This will help to optimize the error correction resources (e.g. which part of the algorithm needs a more expensive error correction). In this paper, we study the error generation and propagation in the HHL algorithm by using a 2-variable SLE. The equations are the same as in [14]. The quantum circuit is implemented and executed in *ibmq_santiago* in the IBM-Q system, which is a 5-superconducting-qubit quantum computer. The results are extracted at each stage of the HHL circuit and compared to the idea solution generated by a MATLAB simulator.

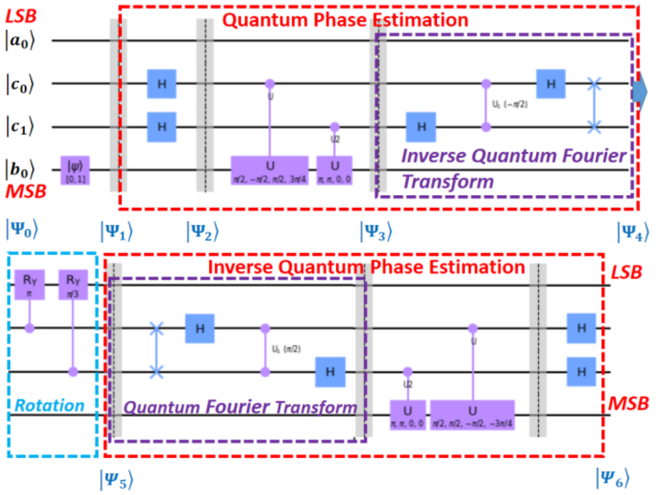

Fig. 1. The HHL quantum circuit implemented in this study. The circuit is partitioned into two parts (top and bottom). The 4-qubit input, $|\Psi_0\rangle$, goes from the left to the right. The output of the top part, $|\Psi_4\rangle$, becomes the input of the bottom part. The output of the HHL circuit is $|\Psi_6\rangle$. Note that the Most-Significant Bit (MSB) is at the bottom.

II. Implementation of HHL Algorithm

The SLE to be solved has $\vec{b}=\begin{pmatrix}0\\1\end{pmatrix}$ and $\boldsymbol{A}=\begin{pmatrix}1 & -1/3\\ -1/3 & 1\end{pmatrix}$ with eigenvalues, λ_1 and λ_2, of 2/3 and 4/3, respectively. The corresponding HHL circuit is shown in Fig. 1. A brief explanation of the HHL algorithm will be provided here but the readers are encouraged to read the step-by-step explanation of the HHL algorithm in [14] and the basics of Quantum Fourier Transform (QFT) and Quantum Phase

*Corresponding Author: hiuyung.wong@sjsu.edu

978-1-6654-9767-1/22/$31.00 ©2022 IEEE

Estimation (QPE) in [12]. We follow the IBM-Q convention [15] to have the Most Significant Bit (MSB) at the bottom of the circuit in Fig. 1. Therefore, the 4 qubits are labeled as $|b_0\rangle|c_1\rangle|c_0\rangle|a_0\rangle$ or $|b_0 c_1 c_0 a_0\rangle$. For example, in a measurement, if the system collapses to $|b_0\rangle = |0\rangle, |c_1\rangle = |1\rangle, |c_0\rangle = |0\rangle, |a_0\rangle = |1\rangle$, it will be labeled as $|0101\rangle$ since $|b_0\rangle$ is the MSB. The input wavefunction is initialized to the ground state with $|\Psi_0\rangle = |0000\rangle$. $|b_0\rangle$ is used to encode $\vec{b}$ and therefore, in this case, a NOT gate is applied to the MSB to change the state to $|\Psi_1\rangle = |1000\rangle$ before the HHL algorithm. The matrix $\boldsymbol{A}$ is encoded in the controlled-U rotations ($\boldsymbol{U} = e^{i\boldsymbol{A}t}$, where t is a parameter). Through QPE, the phases of the eigenvalues of $\boldsymbol{U}$ are obtained and stored in $|c_1\rangle|c_0\rangle$ at $|\Psi_4\rangle$. But the phases of the eigenvalues of $\boldsymbol{U}$ correspond to the eigenvalues of $\boldsymbol{A}$. Therefore, $|c_1\rangle|c_0\rangle$ stores the encoded eigenvalues of $\boldsymbol{A}$. The ancillary qubit $|a_0\rangle$ is then rotated with $|c_1\rangle|c_0\rangle$ as the controlling qubits, and thus is rotated based on the eigenvalues of $\boldsymbol{A}$. The controlled rotation is designed such that the solution of the SLE is encoded as the amplitudes of the basis states [11][14]. At $|\Psi_4\rangle$, the amplitudes are already encoded as the amplitude successfully. However, it does not allow direct measurement due to the entanglement between the qubits. Therefore, inverse QPE is then used to unentangled the qubits and the solution is stored in $|b_0\rangle$ eventually at $|\Psi_6\rangle$.

The solution to this equation is $\vec{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \frac{1}{8}\begin{pmatrix} 3 \\ 9 \end{pmatrix}$ and thus $x_0 : x_1 = 1:3$. Therefore, at the end of the HHL, the is expected that $|b_0\rangle = \frac{1}{\sqrt{10}}(1|0\rangle + 3|1\rangle)$ and the probabilities of measuring $|0\rangle$ and $|1\rangle$ have a ratio of $1^2 : 3^2 = 1:9$.

In a regular HHL circuit, measurement on $|a_0\rangle$ (the LSB) will be performed at $|\Psi_5\rangle$ (i.e. after the rotation of the ancillary bit). The result is discarded if it is $|0\rangle$ and the whole circuit will be recomputed. This is because only when $|a_0\rangle = |1\rangle$ the solution is encoded successfully as the amplitudes. In this study, we only perform this measurement until the end of the circuit ($|\Psi_6\rangle$). This is equivalent to measuring at $|\Psi_5\rangle$ but we need to discard any measurement results that end with LSB = $|0\rangle$.

III. Error Generation and Propagation

We try to understand the error generation and propagation in the HHL circuit in Fig. 1 by running the circuit in IBM-Q [15]. Measurements are performed at $|\Psi_1\rangle$ to $|\Psi_6\rangle$ and compared to the MATLAB simulator [14]. In each measurement, the system will collapse to one of its basis states, namely $|b_0 c_1 c_0 a_0\rangle = |0000\rangle, |0001\rangle, |0010\rangle, \cdots, |1111\rangle$. For convenience, they will be labeled in decimal and named as $|0\rangle, |1\rangle, |2\rangle, \cdots, |15\rangle$. It should be noted that the measurement gives the *square of the magnitude of the amplitude* of each basis in the wavefunction. It is impossible to measure the amplitude in the experiment but the MATLAB simulator can predict the amplitudes.

Fig. 2 shows the simulation and experimental output of $|\Psi_1\rangle$ which is obtained by applying a NOT gate to the MSB of $|\Psi_0\rangle$. Ideally, $|\Psi_1\rangle = |b_0 c_1 c_0 a_0\rangle = |1000\rangle = |8\rangle$, which is what is obtained from the simulator. However, the experiment shows that there is about a 1% of probability obtaining also $|0000\rangle = |0\rangle$, $|1001\rangle = |9\rangle$, $|1010\rangle = |10\rangle$, and $|1100\rangle = |12\rangle$. *It is clear that these are due to the single bit flipping.* For example, $|1001\rangle$ is due to the flipping of *a_0* from the expected outcome $|1000\rangle$. Also, note that the flipping is probably not due to the infidelity of the quantum gate as it is only applied to the MSB (*b_0*). All errors show a similar probability and therefore, the errors are likely due to the noise from the environment (e.g. absorption of a photon). It should be noted that *flipping 2 or more qubits simultaneously is rare*. For example, there is no measurement of $|0001\rangle$ which requires the MSB and LSB to be flipped at the same time.

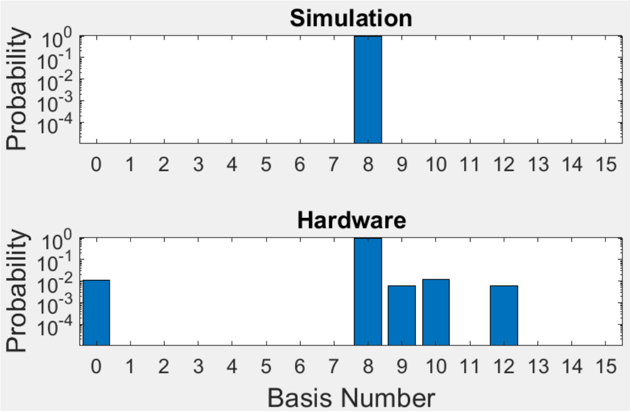

Fig. 2. $|\Psi_1\rangle$ measurement results. Top: simulation. Bottom: Hardware.

Fig. 3 shows the simulation and experimental output of $|\Psi_2\rangle$ which is obtained by applying Hadamard gates to $|c_1\rangle|c_0\rangle$ of $|\Psi_1\rangle$. The Hadamard gate is expected to generate an equal superposition of $|c_1\rangle|c_0\rangle = |0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$ with $|b_0\rangle = |1\rangle, |a_0\rangle = |0\rangle$ and thus the probability of measuring $|8\rangle, |10\rangle, |12\rangle, |14\rangle$ should be the same (0.25). However, the experiment shows that there are unexpected outcomes of $|0\rangle, |2\rangle, |4\rangle, |6\rangle, |9\rangle, |11\rangle, |13\rangle, |15\rangle$. There are two possible sources of error. One is the single-bit flipping error as discussed before. Another is the propagating of errors from $|\Psi_1\rangle$.

If the error is purely due to bit flipping, it is possible to generate all the unexpected outcomes from the expected outcomes (e.g. $|1000\rangle$ with the first bit flipped and becomes $|0000\rangle$). But the probability of the error will be ¼ of that in Fig. 2 because the expected outcomes only have a probability of 0.25 instead of 1. The average probability of the unexpected outcome is about 0.00927 in Fig. 2. Therefore, the probability

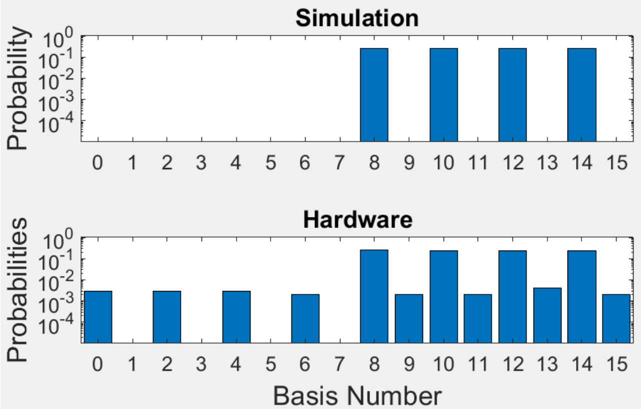

Fig. 3. $|\Psi_2\rangle$ measurement results. Top: simulation. Bottom: Hardware.

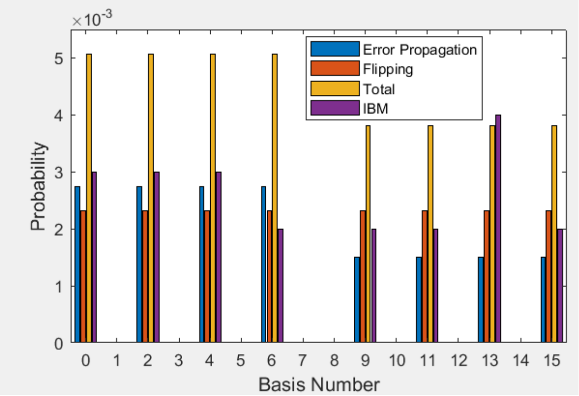

Fig. 4. $|\Psi_2\rangle$ error probabilities of unexpected basis numbers based on various estimations. Only the errors of the unexpected basis states are shown.

of measuring the unexpected outcome due to flipping is expected to be about 0.0023 in $|\Psi_2\rangle$.

If the error is due to the propagation of the error from $|\Psi_1\rangle$, (for example, a $|0000\rangle = |0\rangle$ error component in $|\Psi_1\rangle$ can create $|0\rangle, |2\rangle, |4\rangle, |6\rangle$ in $|\Psi_2\rangle$ due to the Hadamard gates), we can obtain the error in $|\Psi_2\rangle$ by multiplying the matrix with the error in $|\Psi_1\rangle$. The matrix elements connecting the error in $|\Psi_1\rangle$ and $|\Psi_2\rangle$ are all 0.25 (squared for probability). It is found that $|0\rangle$ in $|\Psi_1\rangle$ will contribute an error amplitude of 0.00275 to $|0\rangle, |2\rangle, |4\rangle, |6\rangle$ in $|\Psi_2\rangle$ and $|9\rangle$ will contribute 0.0015 to $|9\rangle, |10\rangle, |13\rangle, |15\rangle$ in $|\Psi_2\rangle$. *Note that the probability (squares of amplitudes) are calculated using the square of the matrix as the experimental amplitudes are unknown. It also means that there will be no interference between different basis errors.*

Fig. 4 plots the probabilities of the unexpected basis errors estimated by different methods. "Total" is the sum of flipping and propagation errors by assuming that there is no correlation between flipping error and propagation error. It can be seen that the experimental error (IBM-Q) is much lower than the "Total" in most cases. *Therefore, the flipping error and propagation error partially cancel each other in some cases*. However, this does not happen for $|13\rangle$ and therefore, cancellation is a random process. We also see that the propagation error and flipping error have similar magnitudes.

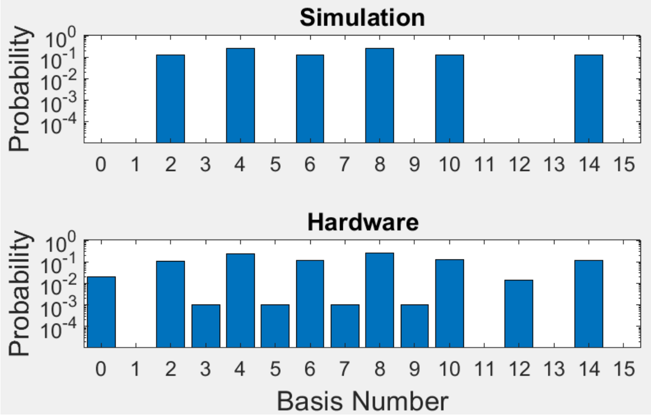

Fig. 5. $|\Psi_3\rangle$ measurement results. Top: simulation. Bottom: Hardware.

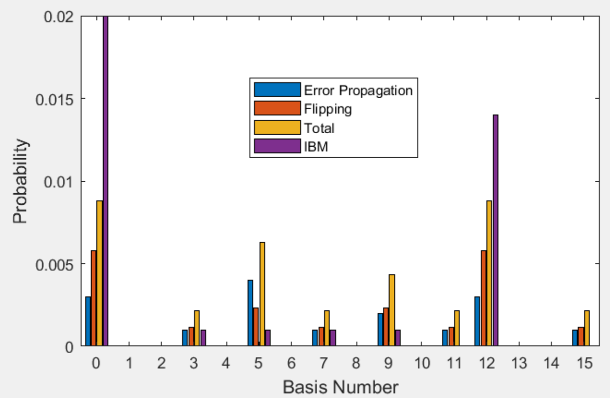

Fig. 6. $|\Psi_3\rangle$ error probabilities of unexpected basis numbers based on various estimations. Only the errors of the unexpected basis states are shown.

Fig. 5 shows the simulation and experimental output of $|\Psi_3\rangle$ after the controlled-U gate is applied. It can be seen that errors due to $|0\rangle$ (0.02) and $|12\rangle$ (0.14) are very high. Compared to the expected outcome such as $|2\rangle$ (0.106), it is almost 20%. *Therefore, the fidelity degrades substantially after this step*. Fig. 6 shows the possible sources of the errors. $|0\rangle$ and $|12\rangle$ can be the results of the flipping errors due to 3 basis states. For example, $|0\rangle = |0000\rangle$ can be obtained by single bit flipping in the expected outcomes $|0010\rangle$, $|0100\rangle$, and $|1000\rangle$. The noise propagated from the previous stage is also higher in $|0\rangle$ and $|12\rangle$. Therefore, the largest errors occur at $|0\rangle$ and $|12\rangle$. However, unlike other errors and in the case of $|\Psi_2\rangle$ where the "IBM-Q" errors are smaller than the sum of the maximum possible flipping errors and propagated errors ("Total"), the "IBM-Q" errors are higher than "Total". This means there is another source of error. Upon further inspection, it is found that $|0\rangle$ and $|12\rangle$ are the only two unexpected outcomes that can be obtained by flipping $|b_0\rangle$. For example, $|12\rangle = |1100\rangle$ can be obtained by flipping the MSB of the expected outcome $|4\rangle = |0100\rangle$. This stage is after a controlled operation on $|b_0\rangle$. *Therefore, it is believed that the additional error is due to the inaccuracy of the control pulse or enhanced flipping due to the inaccurate control pulse, i.e. the gate error.*

Fig. 7 shows the simulation and experimental output of $|\Psi_4\rangle$ after IQFT. The maximum error (e.g. $|8\rangle$ (0.04)) is still about 20% of the expected outcome (e.g. $|12\rangle$ (0.206)). This is similar to $|\Psi_3\rangle$.

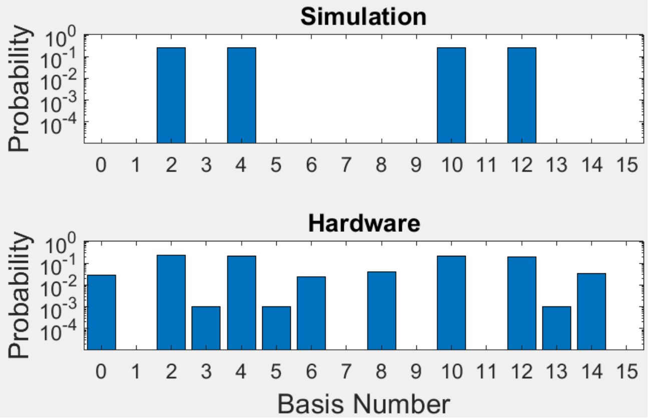

Fig. 7. $|\Psi_4\rangle$ measurement results. Top: simulation. Bottom: Hardware.

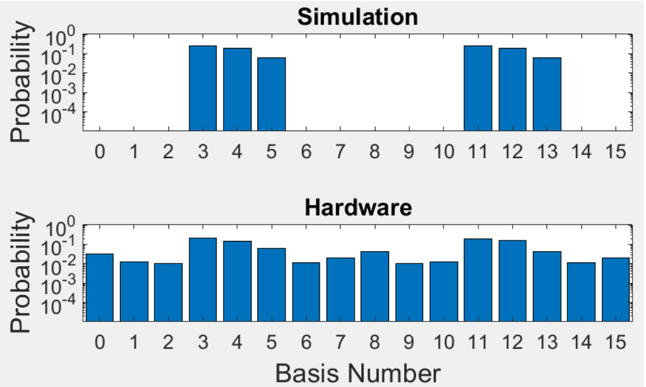

Fig. 8. $|\Psi_5\rangle$ measurement results. Top: simulation. Bottom: Hardware.

However, after the controlled rotation of the ancillary bit $|a_0\rangle$, the error increases substantially as shown in Fig. 8. For example, the unexpected outcome $|8\rangle$ has a probability of 0.042 and the expected outcome $|5\rangle$ has a probability of 0.06. *Therefore, the unexpected outcome is 70% of the expected one.* At $|\Psi_5\rangle$, as explained earlier, the solution is already encoded into the amplitude of the qubits. Although the error is large in a certain basis state (e.g. $|8\rangle$), the overall information is still preserved pretty well. *We argue that if there is a method to extract the information at this stage, the algorithm might work statistically without error correction.*

Unfortunately, for effective information extraction, IQPE is required. Fig. 8 shows the simulation and experimental output of $|\Psi_6\rangle$ after IQPE. As mentioned earlier, in HHL, if the LSB is 0, it will be discarded. Therefore, only odd numbers are meaningful to the results. From the simulation, the probability ratio of $|1\rangle$ to $|9\rangle$ is 0.0625:0.5625 = 1:9 which is what is expected. However, the IBM-Q hardware result is very far from the expected solution. This is due to the error and the error is magnified as the algorithm tries to obtain the solution through interference.

IV. Conclusion

In this paper, we study the error propagation and generation in an HHL quantum algorithm for solving a 2-variable SLE. The 4-qubit quantum circuit is run on a quantum computer based on superconducting qubits. Measurements are performed on 6 parts of the circuits and compared to an ideal MATLAB simulation code. It is found that single-qubit flipping, gate infidelity, and error propagation are the main sources of errors. Two-qubit flipping is rare and not observed. It is found that the information is still pretty well preserved when the solution is obtained after the ancillary qubit rotations (although cannot be extracted efficiently). However, the information is mostly destroyed by the error after the inverse quantum phase estimation which is essential to extract the solution. Therefore, it is suggested that if the error correction resources are limited, they should be placed in the second half of the circuit.

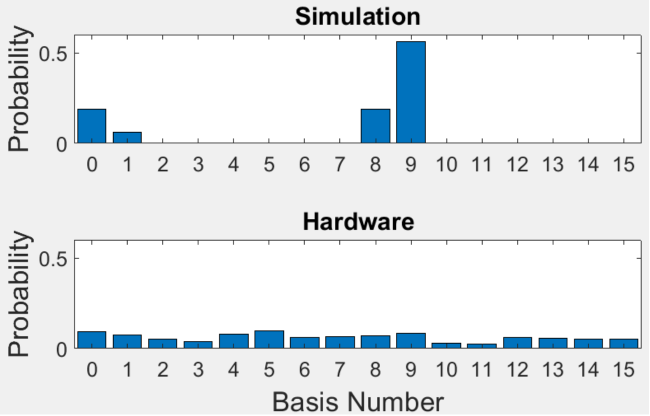

Fig. 9. $|\Psi_6\rangle$ measurement results. Top: simulation. Bottom: Hardware. Note that the y-axis is in a linear scale.

Acknowledgment

Some of the materials are based upon work supported by the National Science Foundation under Grant No. 2046220.

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[2] C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. Deane, "The prospects of quantum computing in computational molecular biology," WIREs Comput. Mol. Sci., 11, e1481 (2021).

[3] D. J. Egger et al., "Quantum Computing for Finance: State-of-the-Art and Future Prospects," in IEEE Transactions on Quantum Engineering, 1, pp. 1–24, 2020, Art no. 3101724, doi: 10.1109/TQE.2020.3030314.

[4] F. Arute, K. Arya, R. Babbush, et al., "Quantum supremacy using a programmable superconducting processor," Nature 574, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5.

[5] A. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. 103, 150502 (2009).

[6] Yudong Cao, Anmer Daskin, Steven Frankel, and Sabre Kais, "Quantum circuit design for solving linear systems of equations," Molecular Physics 110, 15–16 (2011).

[7] R. Chandra, S.C. Eisenstat, and M.H. Schultz, "Conjugate Gradient Methods for Partial Differential Equations," in the Proceedings of the AICA International Symposium on Computer Methods for Partial Differential Equations, Bethlehem, Pennsylvania, June 1975.

[8] S. Dmitry et al., "The Potential of Quantum Computing and Machine Learning to Advance Clinical Research and Change the Practice of Medicine." Missouri medicine 115 (5), 463–467 (2018).

[9] Bojia Duan, Jiabin Yuan, Chao-Hua Yu, Jianbang Huang, and Chang-Yu Hsieh, "A survey on HHL algorithm: From theory to application in quantum machine learning", Physics Letters A 384, 126595 (2020).

[10] Shengbin Wang, Zhimin Wang, Wendong Li, Lixin Fan, Zhiqiang Wei, and Yongjian Gu, "Quantum fast Poisson solver: the algorithm and complete and modular circuit design," Quantum Information Processing 19, Article number: 170 (2020).

[11] H. J. Morrell and H. Y. Wong, "Study of using Quantum Computer to Solve Poisson Equation in Gate Insulators," 2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), 2021, pp. 69-72, doi: 10.1109/SISPAD54002.2021.9592604.

[12] Hiu Yung Wong, Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps. Switzerland: Springer Nature, 2022, pp. 170. https://doi.org/10.1007/978-3-030-98339-0. ISBN-10: 3030983382.

[13] A. G. Fowler, M. Mariantoni, J. M. Martinis and A. N. Cleland, "Surface codes: towards practical large-scale quantum computation," Phys. Rev. A 86, 032324 (2012).

[14] Hector Morrell, Anika Zaman, and Hiu Yung Wong, "Step-by-Step HHL Algorithm Walkthrough to Enhance the Understanding of Critical Quantum Computing Concepts," arXiv:2108.09004v3.

[15] https://quantum-computing.ibm.com/