2024 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD) | 979-8-3315-1635-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/SISPAD62626.2024.10732984

Study of Using Variational Quantum Linear Solver for Solving Poisson Equation

Nithin Reddy Govindugari
Electrical Engineering Department
San *Jose State University*
San Jose, CA, USA
nithinreddy.govindugari@sjsu.edu

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

Abstract— Quantum computing is promising in speeding up the system of linear equations (SLE) solving process. However, its performance is limited by noise. The variational quantum linear solver (VQLS) algorithm is expected to be more resilient to noise than gate-based quantum computing algorithms. This is because error correction is not available yet and VQLS is based on cost function minimization. In this paper, the gate insulator Poisson equation is solved using VQLS. The results are compared to technology computer-aided design (TCAD) results and gate-based quantum algorithm results. We show that, even without error-free qubits, the IBM-Q quantum computer hardware can solve a 2-variable SLE with high fidelity. We further demonstrate that, through VQLS simulation, an 8-variable SLE can be solved with fidelity as high as 0.96.

Keywords—Poisson Equation, Quantum Computing, TCAD, VQLS, Variational Circuit

I. Introduction

Quantum computing (QC) is becoming more promising and quantum supremacy has been demonstrated in a 53-qubit QC system [1]. One of the promising applications of QC is to speed up the solving of the system of linear equations (SLE), $A\vec{x} = \vec{b}$, in which vector $\vec{x}$ is solved for a given matrix, A, and a vector, $\vec{b}$. Harrow-Hassidim-Lloyd (HHL) quantum algorithm was proposed to find the solution exponentially faster than the classical method [2]-[4]. For a give precision, ε, the HHL algorithm scales polynomially with $\log_2 N$ and κ, where N and κ are the size and condition number of the problem, respectively. However, the HHL algorithm is a gate-based model that is very sensitive to noise and error [5]. Since error-corrected quantum computers are not available yet, one cannot even solve a simple 2-variable SLE using the HHL algorithm. For example, in [6], the HHL algorithm is used to solve the Poisson equation across a gate insulator. The HHL algorithm fails to give correct results with the current quantum hardware due to the fact that the noise destroys the quantum interference used in the algorithm [5][7].

Therefore, in the noisy-intermediate-scale-quantum (NISQ) era where error correction is not available, non-gate-based models are more promising. For example, adiabatic quantum computing (AQC) has been used to solve SLE up to 8 variables experimentally on a nuclear magnetic resonance quantum computer [8]. Another approach is the variational quantum linear solver (VQLS) algorithm [9]. VQLS is similar and based on variational quantum eigensolver (VQE) [10]. It has been implemented on a Rigetti superconducting qubit platform for a 1024-variable SLE [9]. However, the fidelity is not shown while

*Corresponding author: hiuyung.wong@sjsu.edu

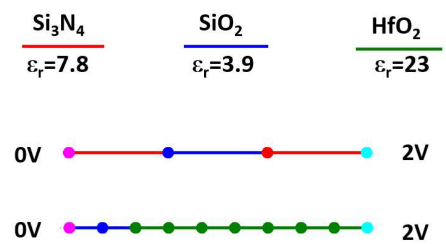

Figure 1: Gate stacks studied in this paper. The top one is dubbed SSS. The bottom one is dubbed HS. Mesh points are shown in dots. Color legend of each material is shown on top with its relative dielectric constant.

it is demonstrated that the cost functions reduce similarly in simulation and experiment. The example in [9] is also based on a special Ising-inspired matrix, A. Therefore, it is worth exploring the performance of VQLS with a technology computer-aided-design (TCAD) problem.

In this paper, using an open superconducting qubit platform [11][12], we study the performance of VQLS in solving the Poisson equation across gate insulators with 2 and 8 variables and compare it to the results using the HHL algorithm.

II. Problems to be Solved

Poisson equations across two 1-D gate stacks are studied. Fig. 1 shows the structures simulated. The first device is a $Si_3N_4/SiO_2/Si_3N_4$ stack (dubbed SSS) and the second device is a SiO_2/HfO_2 structure (dubbed SH). Both structures are biased at 2V and are 2nm thick. The Poisson equation is discretized and the size of A is 2×2 for SSS and 8×8 for SH because the boundary nodes need not be solved. As a result, they can be handled by 1 and 3 qubits respectively in quantum computers. The equations are solved in Python 3.10.4 directly, TCAD Sentaurus using Newton iteration, and Qiskit for quantum computing circuit simulation. SSS is also solved in the *ibm_lago* quantum computer.

III. Variational Quantum Linear Solver (VQLS)

VQLS holds great potential for overcoming the computational complexity limits of classical computers. Fig. 2 shows a block diagram of how VQLS works when solving $A\vec{x} = \vec{b}$. Firstly, $\vec{b}$ is encoded as $|b\rangle$ through amplitude encoding. This is achieved by applying a unitary gate to the ground state $|0\rangle$. For example, in the SSS problem, $A = \begin{pmatrix} 1 & -1/3 \\ -1/3 & 1 \end{pmatrix}$ and

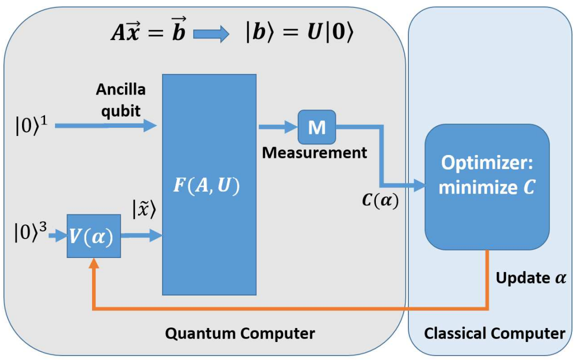

Figure 2: Schematic showing the workflow of VQLS. A 3-qubit case is shown with one ancilla qubit at the top. This corresponds to the SH problem. One of the components of $F(A, U)$ is shown in Fig. 3.

$\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, then $|b\rangle = U|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Note that this is just for finding U to be used in the quantum circuit and $U|0\rangle$ is not a part of the quantum circuit. For the SH problem, $|b\rangle$ has 3 qubits and is an 8-D vector. $|x\rangle$ is obtained through a variational quantum circuit, $V(\alpha)$, applied to the ground state $|0\rangle$. $V(\alpha)$ is also called the ansatz. A variational quantum circuit is a parametrized quantum circuit. It contains rotation gates (e.g, $R_x(\theta)$, $R_y(\theta)$) and entanglement gates. The rotation angles are used as the parameters. These parameters are denoted as α in the figure. Therefore, the goal is to find α such that $A|x\rangle = |b\rangle$ with $|x\rangle = V(\alpha)\,|0\rangle$. Random values are chosen for α as the initial values.

When the solution is found, $A|x\rangle$ and $|b\rangle$ should have the maximum overlap. That means $\langle b|A|x\rangle$ is maximum. Therefore, a cost function that depends on α, $C(\alpha)$, is defined based on minimizing the trace distance between the subspace normalized to $|b\rangle$, i.e. $I - |b\rangle\langle b|$, and $A|x\rangle$, i.e. $A|x\rangle\langle x|A^\dagger$,

$$C_G(\alpha) = \frac{Tr\{[A|x\rangle\langle x|A^\dagger][I-|b\rangle\langle b|]\}}{\langle x|A^\dagger A|x\rangle}, \tag{1}$$

where normalization is performed by dividing the distance by $\langle x|A^\dagger A|x\rangle$. It can be derived that A can be decomposed into a linear combination of L unitary matrices A_l with complex coefficients c_l,

$$A = \sum_{l=1}^{L} c_l A_l. \tag{2}$$

For example, for the SSS problem, $A = 1 \times I - \frac{1}{3} \times X$, where I and X are the identity gate and Pauli-X gate (NOT gate), respectively. Finally, the cost function $C(\alpha)$ is derived to be,

$$C_G(\alpha) = 1 - \frac{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U|0\rangle\langle 0|U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}, \tag{3}$$

which is called the global cost function [9]. However, it is shown that the global cost function can result in Barren Plateaus easily in [9]. Therefore, a local cost function is used instead,

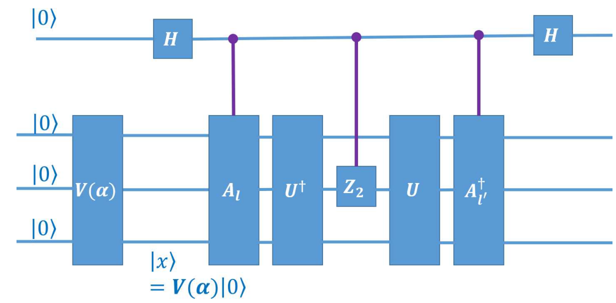

Figure 3: The Hadamard test circuit (without measurement) for computing the real part of $\langle 0|V^\dagger A_{l'}^\dagger U Z_2 \otimes I_{\bar{2}} U^\dagger A_l V|0\rangle$. It can be regarded as the test circuit for computing the expectation value of the operator $A_{l'}^\dagger U Z_2 \otimes I_{\bar{2}} U^\dagger A_l$ for state $V|0\rangle$. Note that no controlled operation is needed for U and $U^\dagger$ because it does not affect the results. The qubits are numbered from 0 to 3 from the top. Therefore, the top qubit is qubit 0. This represents one of the $F(A, U)$ components in Fig. 2.

$$C_L(\alpha) = 1 - \frac{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U\left(\frac{1}{n}\sum_{j=0}^{n-1}|0_j\rangle\langle 0_j|\otimes I_{\bar{j}}\right)U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}, \tag{4}$$

where $|0\rangle\langle 0|$ in the global cost function is replaced by $\frac{1}{n}\sum_{j=0}^{n-1}|0_j\rangle\langle 0_j| \otimes I_{\bar{j}}$. $I_{\bar{j}}$ means that the identity matrix is used for qubits other than qubit j (instead of using the outer product of the ground state of that qubit). To understand this, we should note that $U|0\rangle\langle 0|U^\dagger = |b\rangle\langle b|$. Therefore, by changing to the new form, $|b\rangle$ is not calculated "globally". Instead, each qubit of the ground state receives the action of U "individually" ("locally").

The cost function can be computationally intensive using classical computers and QC is expected to provide tremendous speedup. It can be appreciated that the linear decomposition of A enables the direct computation of the cost function through the Hadamard test [13]. The Hadamard test is used to compute the expectation value of an operator for a given state vector. To implement the Hadamard test, we recognize that $|0_j\rangle\langle 0_j| \otimes I_{\bar{j}} = \frac{I}{2} + \frac{Z_j \otimes I_{\bar{j}}}{2}$. Therefore, to compute $C_L(\alpha)$, we need to compute all terms in $\langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle$ and $\langle 0|V^\dagger A_{l'}^\dagger U Z_j \otimes I_{\bar{j}} U^\dagger A_l V|0\rangle$. Fig. 3 shows the Hadamard test circuit for computing the real part of $\langle 0|V^\dagger A_{l'}^\dagger U Z_2 \otimes I_{\bar{2}} U^\dagger A_l V|0\rangle$ as an example.

The cost is then passed to a classical computer to perform optimization to choose the next set of α for the ansatz. This continues until the cost is less than a pre-set threshold and $V(\alpha_{final})\,|0\rangle$ gives the solution, $|x\rangle$.

There is also an ancillary qubit required to perform the control operation in the Hadamard test (Fig. 2 and Fig. 3). Therefore, for the SSS problem, 2 qubits are needed and for the SH problem, 4 qubits are needed.

IV. Results

Firstly, the SSS problem is run through a VQLS simulator with error emulation and it gives a very high accuracy even with only 15 optimization steps (Fig. 4). The fidelity, which is calculated as the inner product of the true $\vec{x}$ and the computed $\vec{x}$,

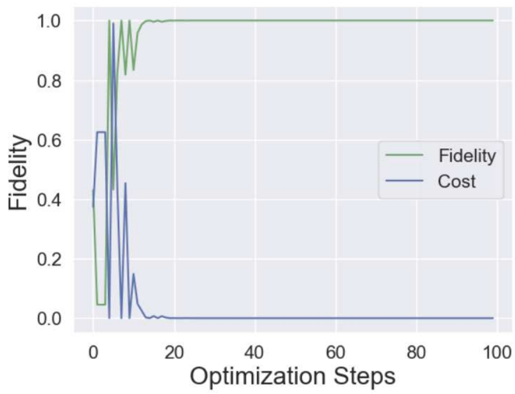

Figure 4: Fidelity and optimization cost of the SSS problem as a function of the optimization step using the VQSL simulator implemented with Qiskit.

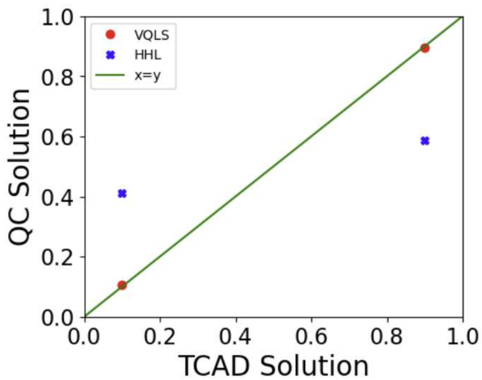

Figure 5: TCAD solution and quantum computing hardware solutions (HHL and VQLS) at the 2 middle nodes of the SSS structure. The squares of the normalized vectors are compared (which represent measurement probabilities in a quantum computer).

is 1 with a precision of 15 digits after the decimal. Since it only has one qubit, no entanglement can be formed in the ansatz. Therefore, the ansatz contains only rotation gates. In Fig. 4, three $R_y(\alpha = \theta)$ are used as the ansatz. Optimizer COBYLA, which stands for "constrained optimization by linear approximation" is used as the optimizer. COBYLA is a numerical optimization method for constrained problems where the derivative of the

Figure 6: Fidelity as a function of optimization step when solving the SSS problem using *ibm_lago* quantum computer.

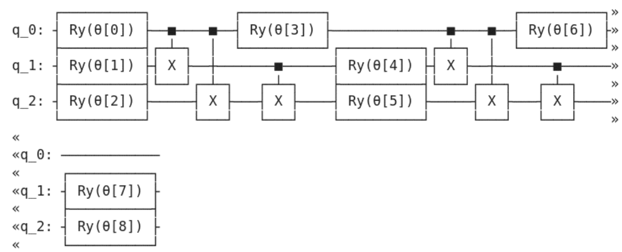

Figure 7: The ansatz used in the SH problem. Rotation gates about the y-axis and controlled-NOT gates as the entanglement gates are used. The rotation gates are parameterized with rotation angles. The initial values are shown. The circuit goes from left to right on the top row and then the second row.

objective function is not known. SparsePauliOp is used to decompose A and Qiskit Estimator is used. It is then run using IBM quantum computer *ibm_lago*. Fig. 5 compares the results from TCAD, quantum hardware using HHL, and quantum hardware using VQLS. It can be seen that while HHL has a solution with a large error, VQLS gives a very similar result as TCAD. Fig. 6 shows that it can be completed in less than 20 steps to reach a fidelity of 99% with the hardware. More importantly, it is stabilized within a similar number of steps as the simulation, which agrees with the observation in [9].

Due to limitations in the access to IBM hardware, structure SH is studied only with simulation with error emulation in Qiskit [12]. Since the hardware result in the SSS case gives a similar result as the VQLS simulation, it is believed that the simulation of VQLS represents the hardware behavior well. Fig. 7 shows the ansatz used in this problem. Entanglement between each pair of qubits is applied through the CNOT gates. Fig. 8 shows the matrix decomposition of A, which is decomposed into the linear combination of 16 unitary gates formed by the tensor products of Pauli matrices and identity gate.

Simulations are conducted many times with different random seeds and Fig. 9 shows three of the typical results. Firstly, most of the time the results are not satisfactory (e.g. Run

```
A = np.array(
[[ 0.16956522, -0.08478261,  0,      0.,    0.,    0.,    0.,     0.  ],
 [-0.08478261,  0.58478261, -0.5,    0.,    0.,    0.,    0.,     0.  ],
 [ 0.,         -0.5,         1.,    -0.5,   0.,    0.,    0.,     0.  ],
 [ 0.,          0.,         -0.5,    1.,   -0.5,   0.,    0.,     0.  ],
 [ 0.,          0.,          0.,    -0.5,   1.,   -0.5,   0.,     0.  ],
 [ 0.,          0.,          0.,     0.,   -0.5,   1.,   -0.5,    0.  ],
 [ 0.,          0.,          0.,     0.,    0.,   -0.5,   1.,    -0.5 ],
 [ 0.,          0.,          0.,     0.,    0.,    0.,   -0.5,    1.  ]]
)
```

'III'	'IIX'	'IIZ'	'IXX'	'IYY'	'IZI'
0.844	-0.396	-0.052	-0.25	-0.25	-0.156
'IZX'	'IZZ'	'XXX'	'XYY'	'YXY'	'YYX'
0.104	-0.052	-0.125	0.125	-0.125	-0.125
'ZII'	'ZIX'	'ZIZ'	'ZZI'	'ZZX'	'ZZZ'
-0.156	0.104	-0.052	-0.156	0.104	-0.052

Figure 8: Matrix $\boldsymbol{A}$ in the SH problem in Python code (top). It is decomposed into 18 unitary gates, A_l, (bottom) with the corresponding coefficients, c_l, shown.

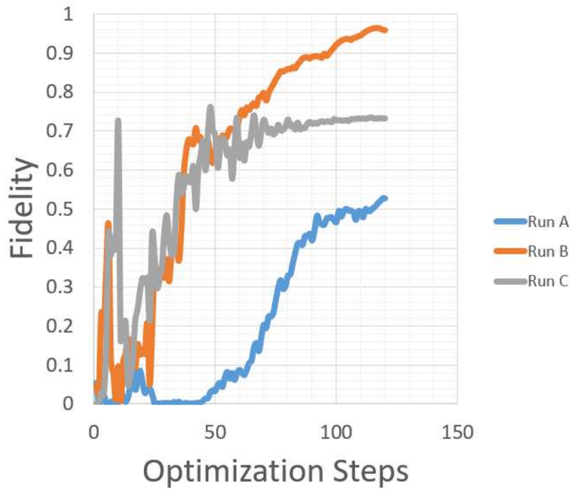

Figure 9: Fidelity as a function of optimization step when solving the SH problem using VQLS simulation in different runs.

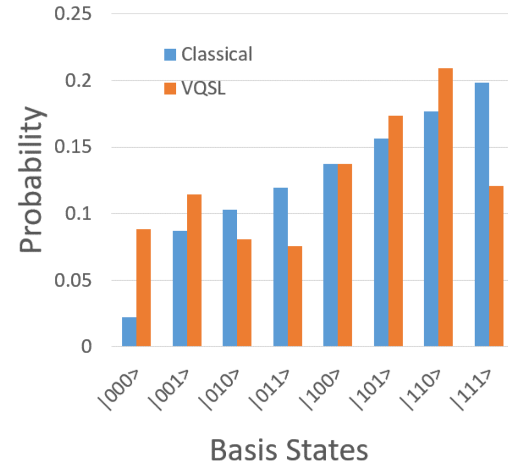

Figure 10: Measurement probabilities of each basis state calculated using VQLS and by squaring the classical solution.

A) which has very low fidelity even with 120 steps. Secondly, sometimes the fidelity plateaus and cannot be further improved (e.g. Run C). The fact that the fidelity of Run C does not further improve after ~70 steps might be due to Barren plateaus [14] where the solution is stuck at a local minimum. Usually, this requires more engineering in the construction of the ansatz because Barren plateaus usually occur in large problems when there are too many entanglements. However, in this case, it appears that one can escape the plateaus with a different random seed (Run B). Run B has achieved the highest fidelity of 96%.

Fig. 10 compares the squares of the components of $\vec{x}$ obtained by TCAD and VQLS. It can be seen that VQLS gives a satisfactory result compared to HHL considering that the qubits are not error-corrected and this is an 8-variable problem. However, the result is still not enough for TCAD applications. This also shows that fidelity alone is not an adequate metrics to determine if the solution of a QC is good enough for TCAD applications. This is consistent with the finding in [6].

V. Conclusions

In this paper, VQLS is used to solve the Poisson equation on two gate stacks, namely a $Si_3N_4/SiO_2/Si_3N_4$ stack (dubbed SSS) and a SiO_2/HfO_2 stack (dubbed SH). They represent a 2-variable and 8-variable system of linear equations, respectively. We used IBM quantum computing hardware to demonstrate that VQLS is resilient to errors in unprotected qubits and obtained 99% fidelity for the SSS problem. This is much better than the HHL algorithm. We then also performed VQLS simulation and were able to obtain 96% of fidelity. This shows that VQLS is promising although Barren Plateaus need to be avoided in large-scale problems.

Acknowledgment

Part of the work was supported by SJSU RSCA Seed Grant Program 2024.

References

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

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

[3] Yudong Cao et al. "Quantum circuit design for solving linear systems of equations," Molecular Physics 110.15-16 (2012), pp. 1675–1680. arXiv: arXiv:1110.2232v2.

[4] A. Zaman, Hector Morrell, and Hiu Yung Wong, "A Step-by-Step HHL Algorithm Walkthrough to Enhance Understanding of Critical Quantum Computing Concepts," in IEEE Access, 2023. 10.1109/ACCESS.2023.3297658

[5] A. Zaman and H. Y. Wong, "Heuristic Error Analysis Framework for Error Bottleneck Identification in Gate-based Quantum Algorithms," IEEE International Conference on Quantum Computing and Engineering (QCE24).

[6] 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), Dallas, TX, USA, 2021, pp. 69-72.

[7] A. Zaman and H. Y. Wong, "Study of Error Propagation and Generation in Harrow-Hassidim-Lloyd (HHL) Quantum Algorithm," 2022 IEEE Latin American Electron Devices Conference (LAEDC), Cancun, Mexico, 2022, pp. 1-4, doi: 10.1109/LAEDC54796.2022.9908231.

[8] Jingwei Wen, et al., "Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing," Phys. Rev. A, 99, 012320, 2019. doi: 10.1103/PhysRevA.99.012320.

[9] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational Quantum Linear Solver," arXiv:1909.05820v4, 2023.

[10] A. Peruzzo et al., "A variational eigenvalue solver on a photonic quantum processor," Nat Commun 5, 4213 (2014). https://doi.org/10.1038/ncomms5213.

[11] IBM Quantum Computing. [Online]. Available: https://www.ibm.com/quantum.

[12] Qiskit Runtime. [Online]. Available: https://cloud.ibm.com/catalog/services/qiskit-runtime

[13] Hadamard test. [Online]. Available: https://en.wikipedia.org/wiki/Hadamard_test

[14] A. Kulshrestha and I. Safro, "BEINIT: Avoiding Barren Plateaus in Variational Quantum Algorithms," arXiv:2204.13751, 2022.