Using IBM-Q to study and visualize the ground state properties of the Su-Schrieffer-Heeger model

Aditya Samaroo and Michael McGuigan
Department of Physics, Adelphi University, Garden City, New York 11530
Computational Science Initiative, Brookhaven National Laboratory, Upton, New York 11973
(Dated: July 24, 2018)

IBM has developed an API allowing users to build and execute quantum programs on their quantum computers built using superconducting circuits, ranging from 4 to 20 qubits. Using the SSH model as our system to study, we compare the accuracy of the ground state found classically to the ground state calculated by the variational quantum eigensolver. We also explore how varying parameters such as number of qubits, quantum depth and shots affected the accuracy of the algorithm.

I. INTRODUCTION

A. Quantum Computing

Quantum computing takes advantage of quantum mechanical properties such as superposition and entanglement in order to perform calculations that are too intensive for a classical computer. Richard Feynmann came up with the idea of quantum computers in 1982, where it was only theoretically motivated until the creation of Shor's algorithm. This algorithm involves the factorization of large numbers, which in turn makes it possible to break encryption keys. Quantum computers would be capable of using Shor's algorithm to crack encryptions at a significantly faster speed than a classical computer, which provided the motivation to start developing quantum computers to run Shor's algorithm and discover other uses. In the present day, companies such as Google, Microsoft, D-Wave Systems, Rigetti Computing, and IBM are building quantum computers using different techniques and racing to achieve quantum supremacy.

The purpose of quantum computing is to perform calculations that are too difficult to be executed on classical computers. But before we are able to use these computers, we must first prove that they are able to use quantum mechanics that will in turn be used to perform the calculations. As a proof of principle we use algorithms that can be done classically and compare those results to that of the quantum computer. In theory, if the quantum computer works for small qubit systems, larger qubit systems can be implemented and run on quantum computers.

B. QISKit

QISKit is IBM's application program interface (API) that allows users to submit their quantum programs to a queue using a credit system. The programs can be run on various quantum computers, ranging from 5 to 20 qubits or the custom quantum simulators. The QISKit library is written in Python and for our project we used the optimization tools, namely the variational quantum eigensolver (VQE). The VQE uses the variational method to reach the lowest energy state (the ground state) of a quantum system. In order to use this eigensolver, we needed to map our quantum system to 2×2 matrices so they represent qubits [1].

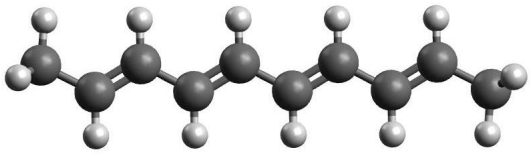

FIG. 1. 3D Model of polyacetylene

C. Su-Schrieffer-Heeger Model

The Su-Schrieffer-Heeger (SSH) Model is a finite 1-dimensional lattice model used to describe polyacetylene $(C_2H_2)_n$ molecules [2–5]. This chain is used to study the effects of topology in condensed matter physics as well as in quantum field theory, where properties of the molecule depend directly on the arrangement and structure. Figure 1 is a 3 dimensional model of polyacetylene created in Avogadro. An interesting trait of this model is that it alternates between single and double bonds. The Hamiltonian for a chain with N groups and mass M is:

$$H_{SSH} = -\sum_{i}^{N} t_{i+1,i}(c_{i+1}^{\dagger}c_i + H.c.) + \frac{K}{2}\sum_{i}^{N}(u_{i+1}-u_i)^2 + \frac{M}{2}\sum_{i}^{N}\dot{u}_i^2 \tag{1}$$

where t is the intersite hopping: $t_{i+1,i} = t_0 - g(u_{i+1} - u_i)$, t_0 is the hopping term without vibrations, g is the electron-phonon coupling constant [4]. $c_i^{\dagger}$ and c_i are the fermionic creation and annihilation operators, which obey the anti-commutation relation: $\{c_i, c_j^{\dagger}\} =$

$a_i a_j^\dagger + a_j^\dagger a_i = \delta_{ij}$. K is the spring constant, u_i is the group displacement along the chain and $\dot{u}_i^2$ is the kinetic energy term. For simplicity, we use the Born-Oppenheimer approximation, which assumes that the chain is perfectly combined or dimerized [2]:

$$u_n = (-1)^n u \tag{2}$$

This makes every displacement term even in distance creating a less complex space for our Hamiltonian.

II. METHODS

A. SSH Model

To generate the Hamiltonian for this system, we used MATLAB. The SSH model involved generating both fermions and bosons. To generate the fermionic ladder operators we used the Grassman number and the Z Pauli matrix [6]. Explicitly, the operators are:

$$a_{F_j} = \underbrace{\sigma_3 \otimes \sigma_3 \otimes \ldots \otimes \sigma_3}_{j\text{-1 times}} \otimes \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{3}$$

$$a_{F_j}{}^\dagger = \underbrace{\sigma_3 \otimes \sigma_3 \otimes \ldots \otimes \sigma_3}_{j\text{-1 times}} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{4}$$

where j identifies which fermion it is. For example, if there were 3 fermions, the first one would tensored with two 2×2 identity matrices, the second one will be tensored with one σ_z matrix and one identity matrix while the third will be tensored with two σ_z matrices. These ladder operators fulfill the anti-commutation relations. Similar to the fermions, bosons are built using their ladder operators. Creating multiple bosons requires them to be tensored with 2×2 identity matrices [7].

$$a_{Bj} = \mathbf{1} \otimes \ldots \otimes \mathbf{1} \otimes \begin{pmatrix} 0 & \sqrt{1} & 0 & \ldots & 0 \\ 0 & 0 & \sqrt{2} & \ldots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \sqrt{n} \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{5}$$

$$a_{Bj}{}^\dagger = \mathbf{1} \otimes \ldots \otimes \mathbf{1} \otimes \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ \sqrt{1} & 0 & 0 & \ldots & 0 \\ 0 & \sqrt{2} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \sqrt{n} & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{6}$$

For multiple bosons, they must fulfill the commutation relation:

$$\left[a_i, a_j{}^\dagger\right] \equiv a_i a_j{}^\dagger - a_j{}^\dagger a_i = \delta_{ij} \tag{7}$$

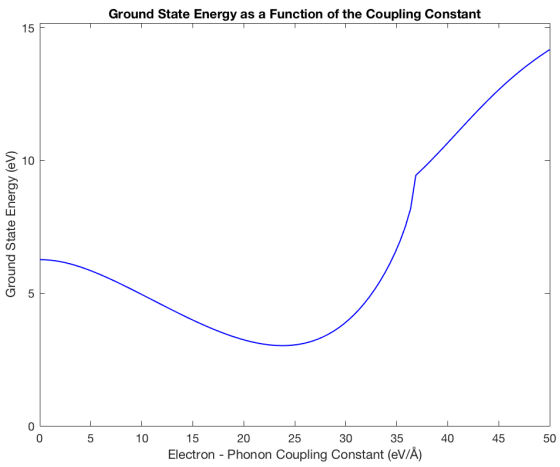

FIG. 2. This graph shows how the coupling constant affects the ground state energy of the SSH Model

$$\left[a_i{}^\dagger, a_j{}^\dagger\right] = \left[a_i, a_j\right] = 0 \tag{8}$$

Since bosons are infinitely dimensional matrices, we must approximate them by using defined matrices. However the only constraint is that the boson must be a power of two so that it can be mapped onto multiple qubits if it were larger than a 2×2 matrix.

The coupling constant g is how the electrons and phonons interact in the SSH model. To see how the coupling affected the ground state energy of our system, we used MATLAB to visualize the ground state as a function of coupling (Figure 2).

B. Variational Quantum Eigensolver (VQE)

In order for the variational eigensolver to work, various libraries must be imported (plotting, qiskit, numpy, scipy, functools). You input the number of qubits you are using for the Hamiltonian (in the case of my Hamiltonian, 2^n plus the size of boson matrix), as well as the quantum depth. The quantum depth is how many times the same circuit is being used per trial. Shots is defined as how many times the program will calculate the energy then average, if the amount of shots is 1, then the average is exactly only one execution for the trial. The program then imports the list of coefficients corresponding to the tensored Pauli matrices. The list is then read and the Hamiltonian is generated.

Quantum circuits are a sequence of quantum gates where each gate can act on a qubit in different ways. Some gates can change the phase of the qubit, the Hadamard gate (H gate) allows the qubit to be in superposition of $|1\rangle$ and $|0\rangle$, meaning that when that qubit is measured, there will be an equal probability of obtaining either state. In the case of the quantum eigensolver, the quantum circuit is your trial state.

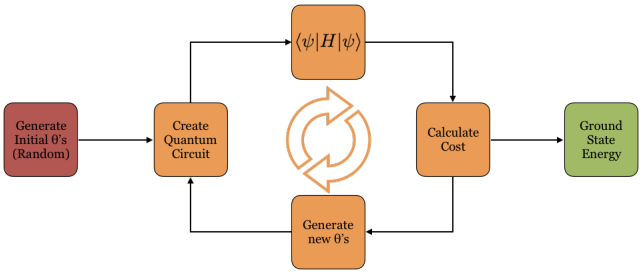

FIG. 3. Flowchart of the VQE. It begins with initial values for your trial state and minimize them in order to obtain the ground state energy.

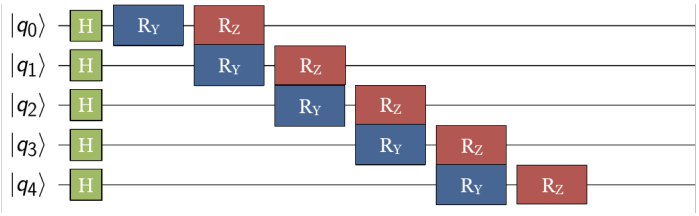

FIG. 4. Representation of a quantum circuit. This circuit consists of arbitrary R_y and R_z rotations to reach the ground state energy.

The variational method says that the expectation value of a Hamiltonian when using trial states will give you an upper bound.

$$\langle\psi_{trial}|\, H\, |\psi_{trial}\rangle \geq E_{ground} \tag{9}$$

This method consists of varying the trial state parameters until the upper bound is minimized. Once the parameters are minimized the expectation value will converge to the ground state energy:

$$\langle\psi_{trial}|\, H\, |\psi_{trial}\rangle = E_{ground} \tag{10}$$

Figure 3 is a flowchart of the VQE. It starts with initial parameters and the quantum circuit is created. The circuit is then evaluated and the lowest energy state is the cost of that expectation value. This process is repeated for a set amount of trials defined by the user and the final cost of the optimization is the ground state energy which the algorithm converges to.

Figure 4 is a representation of the trial circuit used to evaluate our Hamiltonian. Each qubit is first put in a state of superposition, then an R_y and R_z rotation is applied. Once the circuit is evaluated and the next trial begins, new parameters are passed to the circuit which can either increase or decrease the upper bound when the next evaluation occurs.

C. Parameters

To generate the Hamiltonian, the coupling constant g = 4.8 eV/Å, K = 17.3 eV/Å^2 and $t_0 = 1$. Physical constants such as $\hbar$, ω, and mass are set to 1 for simplicity.

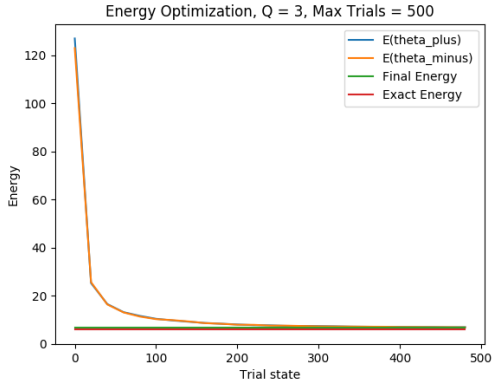

FIG. 5. Energy optimization trial run for a 5 qubit system with a quantum depth of 3. The percent error was 6.7%.

For this simulation, we used 2 fermions (4×4), and an 8×8 boson, resulting in a 32×32 Hamiltonian, which is a 5 qubit system. The quantum depth was varied from 1 to 10, and the amount of shots used were 1 and 100. The default trial circuit from IBM-Q was used to perform the optimization which was done the quantum computer simulator.

III. RESULTS

With a quantum depth of 3, the eigensolver is able to converge to the ground state energy with a 7% error. Figure 5 shows that at a quantum depth of 3 that the algorithm converges to an energy state above the exact ground state of the given system.

Comparing quantum depth to how well the variational quantum eigensolver converges, (Figure 6) after increasing the depth from 2 to 3, there is a significant difference, but after increasing it further, the results did not change. This proves that increasing the quantum depth past a certain point does not improve your optimization.

Simulating a 5 qubit system with shots equal to 100 did not yield good results. The algorithm converged to an energy that was 400% greater than the ground state energy of the system (Figure 7). An explanation to why this calculation did not converge is that there was not enough data for the variational quantum eigensolver to converge properly. If the amount of shots per trial were increased to 2048 or more, the eigensolver will have an easier time converging to the correct ground state. The only pitfall for increasing the amount of shots is the amount of time needed to complete the calculation.

IV. CONCLUSIONS

We studied and visualized the ground state properties of the Su-Schrieffer-Heeger model by creating the Hamiltonian in MATLAB and using the variational quantum

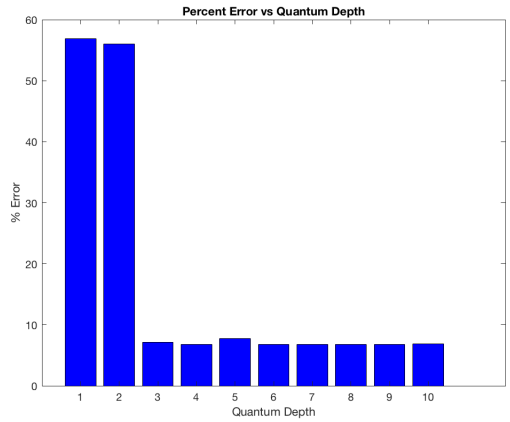

FIG. 6. Quantum depth versus percent error. There is a significant difference when using a depth of 2 versus a depth of 3.

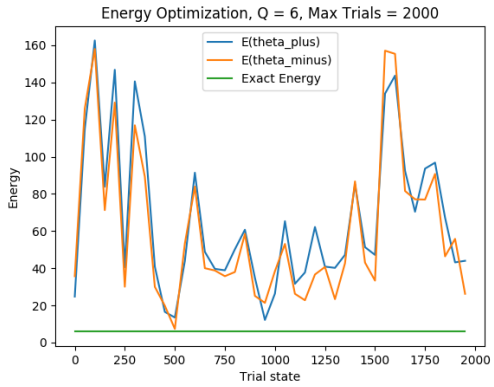

FIG. 7. Energy optimization trial for a 5 qubit system with a quantum depth of 1 and shots equal to 100. The percent error was 399.1%.

eigensolver to calculate the ground state energy on the IBM-Q. The variational quantum eigensolver is able to converge to the ground state energy of our system with a minimum of 7% error. More research is needed into how to create improved trial functions that will allow the eigensolver to converge to the correct ground state. Large systems such as the one we have studied require more complex trial circuits. Increasing the amount of shots can also allow the algorithm to converge closer to the correct ground state, more iterations per trial makes for a better average. Next steps include studying more realistic systems, which require more qubits and tend to be more complex.

V. ACKNOWLEDGEMENTS

This project was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI). I would also like to thank Eugene Dumitrescu from Oak Ridge National Lab for his insight on quantum computing and quantum algorithms.

REFERENCES

[1] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242–246 (2017).
[2] W. P. Su, J. R. Schrieffer, and A. J. Heeger, Physical Review B **28**, 2099–2111 (1983).
[3] J. K. Asbóth, L. Oroszlány, and A. Pályi, A Short Course on Topological Insulators Lecture Notes in Physics , 1 (2016).
[4] J. P. Hague and C. Maccormick, New Journal of Physics **14**, 033019 (2012).
[5] F. L. J. Vos, D. P. Aalberts, and W. V. Saarloos, Physical Review B **53**, 14922–14928 (1996).
[6] P. Woit, Quantum Theory, Groups and Representations , 357–363 (2017).
[7] R. Shankar, *Principles of quantum mechanics* (Springer, 2014).