An Introduction to Quantum Computing and its Applications

Aditya Samaroo

Department of Physics
Adelphi University

May 14, 2018

Abstract

The purpose of this thesis is to introduce the reader to the world of quantum computing. Beginning with quantum harmonic oscillators as well as the basics of quantum gates and building quantum circuits. It will end with a research application of quantum computing and steps to be taken looking forward.

Acknowledgements

I would like to thank my advisor Dr. Matthew Wright for his guidance and support during my career at Adelphi University. I would also like to thank my mentor Dr. Michael McGuigan from Brookhaven National Lab (BNL) for the opportunity to research and experience the world quantum computing, and Dr. Eugene Dumitrescu from Oak Ridge National Lab (ORNL) for his insight and discussions into quantum computing and quantum algorithms.

Contents

Chapter 1

Introduction

The idea that using the properties of quantum mechanics to amplify computational power was first introduced by Richard Feynman during a talk in 1982 [1]. The study of quantum computing was theoretically motivated until the proposal of Shor's algorithm. This algorithm involves finding the prime factors of a number N. This algorithm would be able to break modern day encryptions, and it involves the use of the quantum Fourier transform to achieve polynomial time efficiency [2]. 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 this and other quantum algorithms. 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.

Quantum computers take advantage of quantum mechanical properties such as superposition and entanglement to perform calculations and store more information in a quantum bit (qubit). Qubits are acted on by quantum gates so that they perform the analog actions to classical computers. 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.

Chapter 2

Quantum Harmonic Oscillator

2.1 Bosonic Harmonic Oscillator

The quantum harmonic oscillator is the quantum mechanical analog to the classical harmonic oscillator which describes a system that experiences a restoring force when displaced from its equilibrium point. This system can be solved analytically which is rare in quantum mechanics. The Hamiltonian of the quantum harmonic oscillator is described by:

$$H = \frac{P^2}{2m} + \frac{1}{2}kX^2 = \frac{P^2}{2m} + \frac{1}{2}mw^2X^2 \tag{2.1}$$

where m is the mass of the particle, ω is the angular frequency, X and P are the position and momentum operators respectively [3–6].

The harmonic oscillator can also be described using ladder operators. This method allows for energy eigenvalues to be found without solving the differential equations. The annihilation operator is responsible for lowering the state while its adjoint, the creation operator, raises the state. The annihilation operator and creation operators are defined by:

$$a = \sqrt{\frac{m\omega}{2\hbar}}(X + \frac{i}{m\omega}P) \tag{2.2}$$

$$a^\dagger = \sqrt{\frac{m\omega}{2\hbar}}(X - \frac{i}{m\omega}P) \tag{2.3}$$

These operators can be applied to these energy eigenstates $|n\rangle$ lowering or raising them to another state:

$$a\,|n\rangle = \sqrt{n}\,|n-1\rangle \tag{2.4}$$

$$a^\dagger \left|n\right\rangle = \sqrt{n+1}\left|n+1\right\rangle \tag{2.5}$$

We can write the position and momentum operators in terms of the ladder operators:

$$X = \sqrt{\frac{\hbar}{2m\omega}}(a^\dagger + a) \tag{2.6}$$

$$P = i\sqrt{\frac{\hbar m\omega}{2}}(a^\dagger - a) \tag{2.7}$$

In matrix representation, the ladder operators are described by infinite matrices. However, we can approximate them using finite matrices.

$$a = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \sqrt{n} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \tag{2.8}$$

$$a^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ \sqrt{1} & 0 & 0 & 0 & \dots & 0 \\ 0 & \sqrt{2} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{3} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \sqrt{n} & 0 \end{pmatrix} \tag{2.9}$$

These operators obey the following commutation relations:

$$\left[a_i, {a_j}^\dagger\right] \equiv a_i {a_j}^\dagger - {a_j}^\dagger a_i = \delta_{ij} \tag{2.10}$$

$$\left[{a_i}^\dagger, {a_j}^\dagger\right] = [a_i, a_j] = 0 \tag{2.11}$$

We can describe the Hamiltonian in terms of the ladder operators:

$$H = \hbar\omega\left(a^\dagger a + \frac{1}{2}\right) \tag{2.12}$$

where $a^\dagger a$ is the number operator N.

2.2 Fermionic Harmonic Oscillator

Bosons are infinitely dimensional matrices, which means that they have an infinite amount of energy states. Fermions on the other hand are only 2 × 2 matrices with two energy states. An example of a fermion we can look at is the electron. The two different states are spin up and spin down state. These particles follow Fermi-Dirac statistics as well as obey the Pauli exclusion principle, which states that two identical fermions cannot occupy the same state.

The fermionic harmonic oscillator is similar to the harmonic oscillator except the operators follow the anticommutation relations unlike the "bosonic oscillator" which follows the commutation relations. While the bosonic oscillator is symmetric, the fermionic oscillator is antisymmetric [4, 5]. The ladder operators are the same as the operators in the bosonic oscillator system. The matrix representation however is different:

$$a_F = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{2.13}$$

$$a_F^\dagger = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{2.14}$$

In order to generate j fermion pairs, one must tensor the ladder operators with Pauli matrices and identity matrices to build larger matrices [4].

$$a_{F_j} = \underbrace{\sigma_3 \otimes \sigma_3 \otimes \ldots \otimes \sigma_3}_{j\text{-1 times}} \otimes \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{2.15}$$

$$a_{F_j}{}^\dagger = \underbrace{\sigma_3 \otimes \sigma_3 \otimes \ldots \otimes \sigma_3}_{j\text{-1 times}} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1} \tag{2.16}$$

The fermionic ladder operators follow the anticommutation relations:

$$\left\{a_i, a_j{}^\dagger\right\} \equiv a_i a_j{}^\dagger + a_j a_i{}^\dagger = \delta_{ij} \tag{2.17}$$

$$\left\{a_i{}^\dagger, a_j{}^\dagger\right\} = \{a_i, a_j\} = 0 \tag{2.18}$$

This is necessary to build larger matrices in order to study larger quantum systems as you will see in the following chapters.

Chapter 3

Quantum Computing

3.1 Qubits

A qubit is a logical bit that is in a superposition of 0 and 1. In a quantum mechanical sense, a qubit is in a superposition of two states: $|0\rangle$ and $|1\rangle$. This superposition of states can be written as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3.1}$$

where $|0\rangle = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$. The coefficients α and β represent the probability amplitudes of each state. These coefficients are constrained by: $|\alpha|^2 + |\beta|^2 = 1$. One can represent a qubit using the Bloch sphere. Figure 3.1 is a Bloch sphere with a radius of 1, which shows only three degrees of freedom although coefficients of the states can be complex or imaginary. The fourth degree of freedom is eliminated by the normalization constraint. The choice of having $|0\rangle$ at the top and $|1\rangle$ at the bottom of the sphere is arbitrary.

3.2 Quantum Gates

Quantum gates are the quantum analog to logic gates in classical computers. They perform operations on a qubit and combining quantum gates builds a quantum circuit. Gates can operate on single qubits or multiple qubits. There are 10 commonly used gates are the Hadamard (H) gate, Pauli-X (NOT) gate, Pauli-Y gate, Pauli-Z (R_π) gate, Identity (I) gate, Phase shift

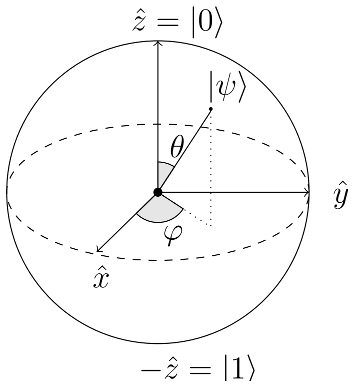

Figure 3.1: A Bloch sphere can be used to illustrate a two level quantum mechanical system, also known as a qubit. The pure state ψ has parameters θ and ϕ which are polar angles in spherical coordinates.

(R_ϕ) gate, SWAP (S) gate, Controlled (cX, cY and cZ) gates, Toffoli (CCNOT) gate, and the Controlled SWAP (cS) gate. These gates can be represented by unitary matrices that have the following properties:

$$U^*U = I \tag{3.2}$$

$$U^\dagger U = UU^\dagger = I \tag{3.3}$$

The Hadamard gate acts on a single qubit, putting it in a superposition of $|0\rangle$ and $|1\rangle$. It maps each state such that they show an equal probability of being measured, generating superposition.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

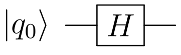

Figure 3.2: On the left is the matrix representation of the Hadamard gate. On the right is the circuit representation of the gate acting on one qubit.

The Pauli-X gate, also known as the NOT gate is a single qubit gate that maps the current state to the opposite state: $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. It is also called the bit-flip gate. On the Bloch sphere it is equivalent to a rotation along the X axis.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

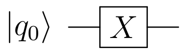

Figure 3.3: On the left is the matrix representation of the Pauli-X gate. The right shows the circuit representation of the gate acting on one qubit.

The Pauli-Y gate acts on a single qubit. It is used to map $|0\rangle$ to $i\,|1\rangle$ and $|1\rangle$ to $-i\,|0\rangle$. On the Bloch sphere it is equivalent to a rotation along the Y axis.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

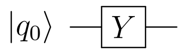

Figure 3.4: The matrix on the left is the Pauli-Y gate. The figure on the right is the circuit representation of the gate acting on one qubit.

The Pauli-Z (R_π) gate acts on a single qubit and applies a rotation along the Z axis. It leaves $|0\rangle$ unchanged while mapping the state $|1\rangle$ to $-\,|1\rangle$.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

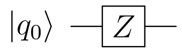

Figure 3.5: The matrix on the left is the Pauli-Z gate. The diagram on the right is the circuit representation of the gate acting on one qubit.

Phase shift (R_ϕ) gate acts on a single qubit and leaves the $|0\rangle$ unchanged and mapping $|1\rangle$ to $e^{i\phi}\,|1\rangle$. On the Bloch sphere this is equivalent to a rotation of ϕ degrees around the sphere.

$$R_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{\phi} \end{bmatrix}$$

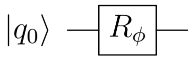

Figure 3.6: On the left is the matrix representation of the phase shift gate. The diagram on the right is the circuit representation of the gate acting on one qubit.

The Identity (I) gate performs an idle operation on the qubit, leaving the states unchanged. For a single qubit, a 2×2 matrix is used to act on it.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

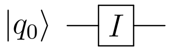

Figure 3.7: The matrix on the left is a 2×2 identity matrix. On the right is the circuit representation of the gate acting on one qubit.

The controlled gates act on 2 or more qubits, where the first qubit is used as a control. The control-X gate is also known as the (CNOT) gate. where it only flips the qubit when the first state is $|1\rangle$. CNOT gates are used to entangle two qubits [7].

Entanglement is another quantum mechanical property that qubits use. When one of the two qubits are measured, the state of the other qubit is also known. Figure 3.8 shows the circuit representation of a CNOT gate where $|q_1\rangle$ is the control qubit and $|q_0\rangle$ is the target. Table 3.1 shows the possible outcomes of the CNOT gate depending on the state of each qubit.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

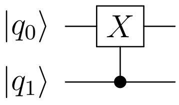

Figure 3.8: Figure is the matrix representation of the CNOT gate. Figure is the circuit representation of the gate, where $|q_1\rangle$ is the control qubit and $|q_0\rangle$ is the target qubit.

Before		After	
Control	Target	Control	Target
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Table 3.1: A table of possible outcomes when two qubits are acted on by a CNOT gate.

The other controlled gates are similar to the controlled-X gate, except it uses the Y or Z gate along the diagonal of the following matrix. It performs the same operation as the original gate but uses a qubit for a control and the other as a target.

$$C(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{10} \\ 0 & 0 & u_{01} & u_{11} \end{bmatrix} \tag{3.4}$$

3.3 IBM Quantum Experience

3.3.1 QISKit

QISKit is an application program interface (API) developed by IBM to create and run programs on their quantum computers and their quantum computer simulators. The scripts are written in Python and submitted to a queue to be ran. Each program requires credits relative to how many quantum executions are being done on the computer.

The available backends available for use are the two 5 qubit (IBM-Q 5 and IBM-Q 5.1) and the 16 qubit (IBM-Q 16) systems. The quantum computer simulator is also available for use, which provides an ideal environment for programs to be calculated. Error rates for these backends are 10^{-3} which provides a reasonable approximation for a majority of the experiments performed.

3.3.2 Mapping Quantum Systems to Qubits

An n qubit system is capable of storing a 2^n size matrix. For example, a 32×32 Hamiltonian can be mapped to a 5 qubit system. In order to map a system to a qubit, they must be factored into Pauli matrices based on the size of the matrix.

$$H = \sum_{i=1}^{n^k} x_i \sigma_i \tag{3.5}$$

Equation 3.5 shows how we can write the Hamiltonian in terms of a permutation of Pauli matrices and 2×2 identity matrix multiplied by some coefficient x. n^k represents the amount of permutations (n is the different 2×2 matrices and k is the number of elements being tensored together) there are depending on the size of the system and σ_i is that permutation of the tensored matrix. We can take the trace of the Hamiltonian and the tensored Pauli matrix combinations to find the coefficients needed to rebuild the system on the quantum computer. To find the coefficient x_i for a q qubit system, we use the following equation:

$$x_i = \frac{tr(H\sigma_i)}{2^q} \tag{3.6}$$

where σ_i represents the i^{th} permutation of the tensor products. Once the nonzero coefficients are found for all possible combinations of Pauli matrices, they can be read into the Python script and rebuilt on the quantum computer.

Chapter 4

Su-Schrieffer-Heeger Model

4.1 Background

The Su-Schrieffer-Heeger (SSH) Model is a finite 1-dimensional lattice model used to describe polyacetylene $(C_2H_2)_n$ molecules [6, 8–19]. 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 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.

This model follows the second quantization of the tight-binding model which is written as:

$$H = -t \sum_{\langle \sigma_i, \sigma_j \rangle} (c^\dagger_{i,\sigma}, c_{j,\sigma} + H.c.) \tag{4.1}$$

where t is the hopping integral, σ is the spin polarization i and j represent the nearest neighbor index, and $H.c.$ is the Hermitian conjugate of the operators. The creation operator $c_i^\dagger$ creates an electron at site i while c_j annihilates an electron at site j. These operators obey the anticommutation relation:$\{c_i, c_j^\dagger\} = a_i a_j^\dagger + a_j^\dagger a_i = \delta_{ij}$

The SSH model is of spinless electrons so the σ term is not relevant [9]. The Hamiltonian for a chain with N groups and mass M is:

$$H_{SSH} = -\sum_i^N t_{i+1,i}(c^\dagger_{i+1} c_i + H.c.) + \frac{K}{2} \sum_i^N (u_{i+1} - u_i)^2 + \frac{M}{2} \sum_i^N \dot{u}_i^2 \tag{4.2}$$

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 (fermion-boson) coupling constant [10]. 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 [8, 20]:

$$u_n = (-1)^n u \tag{4.3}$$

This makes every displacement term even in distance, reducing the complexity of our Hamiltonian.

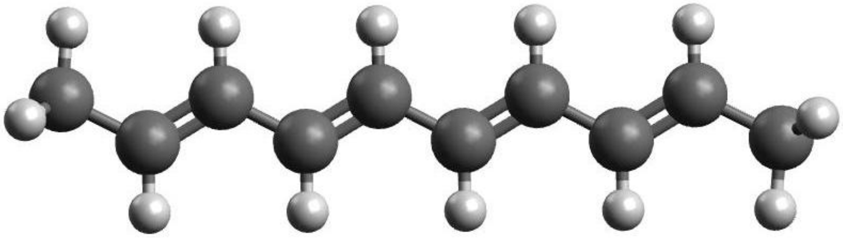

Figure 4.1: A 3D representation of polyacetylene $(C_2H_2)_n$. n represents the how many C_2H_2 groups are on the chain. The chain alternates between single and double bonds.

4.2 Project

4.2.1 Background

The objective of this project is to use IBM's QISKit to find the ground state energy of the SSH model. The variational quantum eigensolver is the ideal method to achieve this. We can vary certain parameters as well as the system itself to see how well the eigensolver does with changing variables [7, 21].

The system is first created and factored in MATLAB before being written to a text file. The text file is then read into a Python script and the variational quantum eigensolver then picks a random starting point and optimizes the parameters until the maximum amount of trials is reached. At the end of the optimization the program should achieve the ground state energy of the system or approach it.

4.2.2 Variational Quantum Eigensolver

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

$$\langle\psi_{trial}| H |\psi_{trial}\rangle \geq E_{ground} \tag{4.4}$$

This method consists of varying the trial state parameters until the upper bound is minimized. These parameters are adjusted until it converges to the ground state energy:

$$\langle\psi_{trial}| H |\psi_{trial}\rangle = E_{ground} \tag{4.5}$$

The variational quantum eigensolver (VQE) 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 [22–29]. 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.

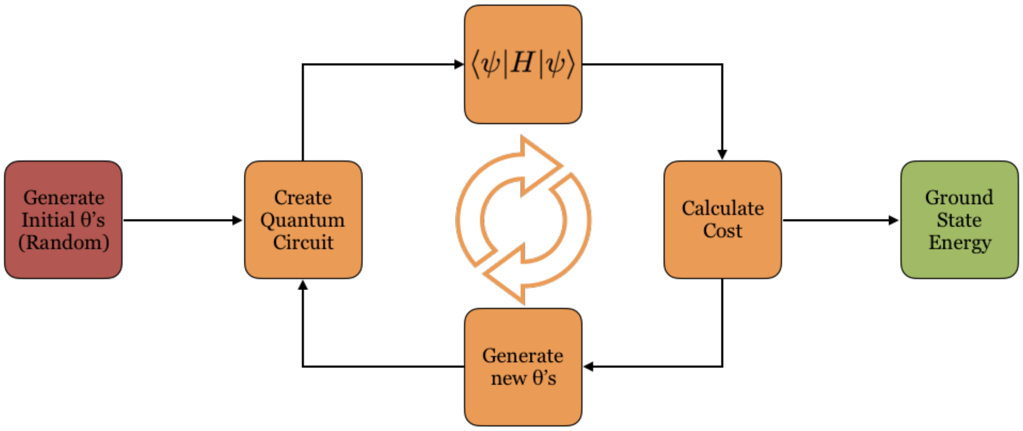

Figure 4.2: Flowchart of the VQE. It begins with initial values for your trial state and minimize them in order to obtain the ground state energy.

Quantum depth is a parameter that can be varied in the eigensolver. The depth refers to the amount of times the quantum circuit is repeated. Circuit depth can help yield better results in some cases such as executing programs on a quantum computer simulator since it does not experience decoherence. Figure 4.3 is a representation of the trial circuit used evaluate

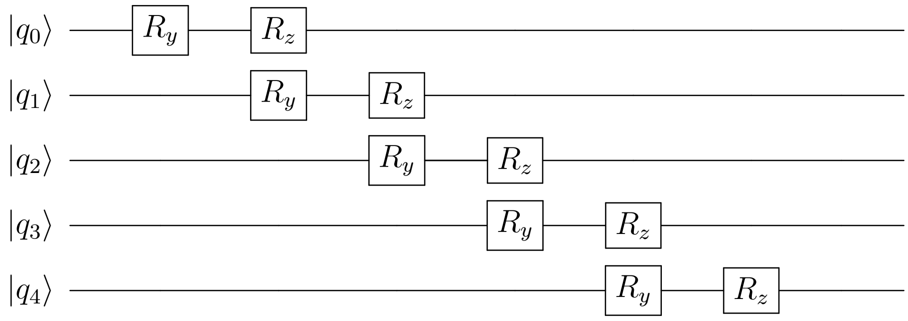

Figure 4.3: Representation of the trial circuit used for the VQE. The depth of the circuit is 1 with an R_y and R_z gate are acting on each qubit with at first an arbitrary rotation. The VQE then optimizes the angles necessary to approximate the ground state energy.

the Hamiltonian. The parameters being optimized are the angles, in radians, used in the gates. By minimizing these parameters, the ground state energy can be achieved.

Quantum circuits are essentially the state that are acting on the system. By adding or removing gates, the state is essentially being changed to either be unsuitable or fit the system. A better trial state consists of following patterns that the Hamiltonian show. By constructing a state that shows a similar pattern to the system, it will converge to the ground state faster than if it did not follow a similar pattern [30].

4.3 Results

With a quantum depth of 3, the eigensolver is able to converge to the ground state energy with a 7% error. Figure 4.4 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.

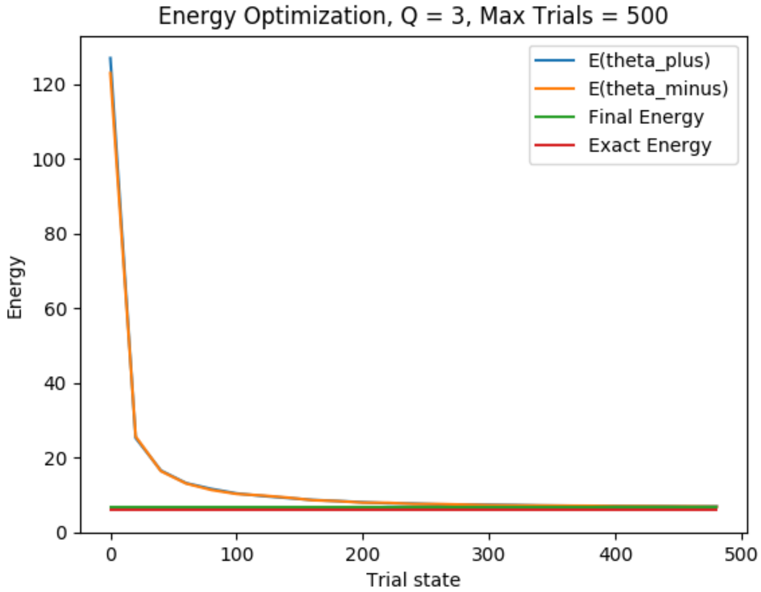

Figure 4.4: Energy optimization trial run for a 5 qubit system with a quantum depth of 3. The percent error was 6.7%.

Comparing quantum depth to how well the variational quantum eigensolver converges, (Figure 4.5) 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.

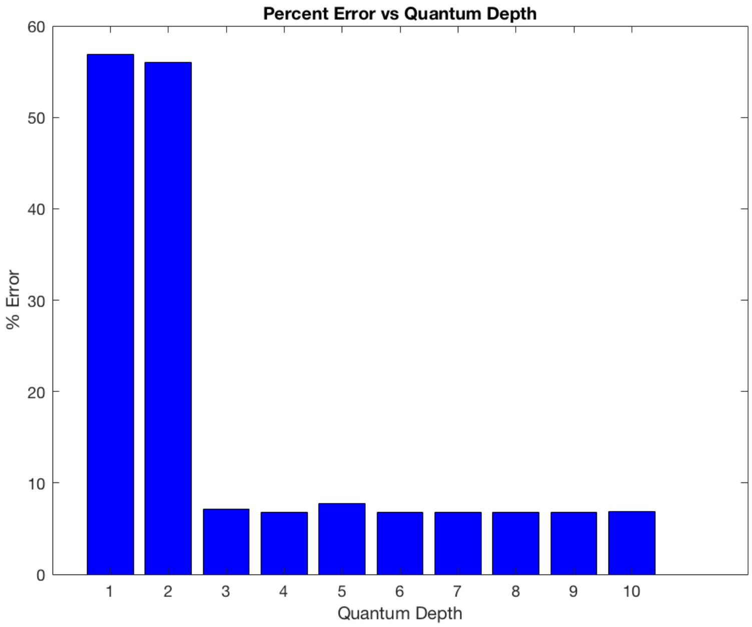

Figure 4.5: Quantum depth versus percent error. There is a significant difference when using a depth of 2 versus a depth of 3.

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 4.6). 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.

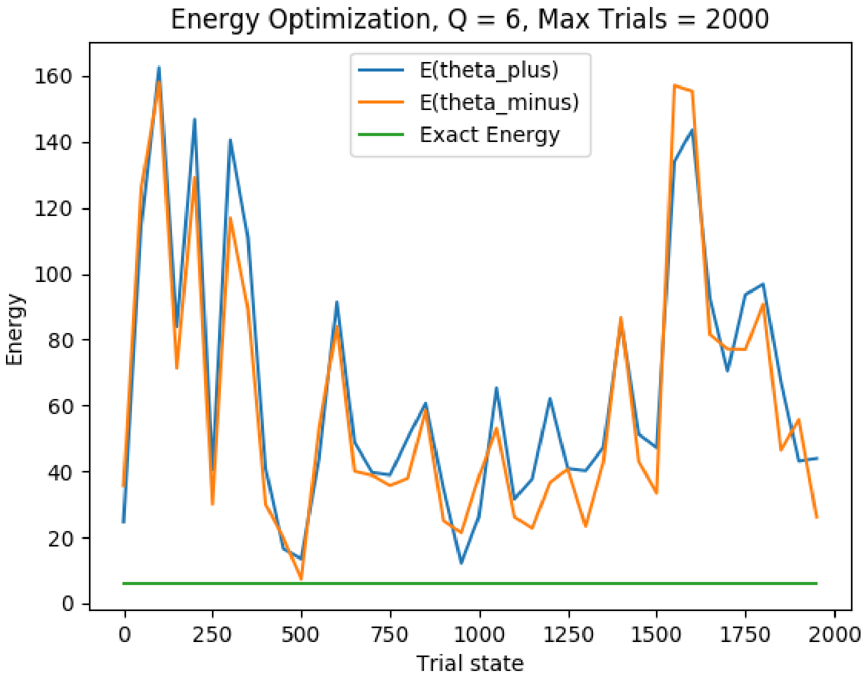

Figure 4.6: Energy optimization trial for a 5 qubit system with a quantum depth of 1 and shots equal to 100. The percent error was 399.1%.

4.4 Conclusion

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 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.

References

[1] R. P. Feynman, “Simulating physics with computers”, International Journal of Theoretical Physics **21**, 467–488 (1982).

[2] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, SIAM Review **41**, 303–332 (1999).

[3] T. Nishioka, “Entanglement entropy: holography and renormalization group”, (2018).

[4] P. Woit, “The fermionic oscillator”, Quantum Theory, Groups and Representations, 357–363 (2017).

[5] A. Das, “Supersymmetry and finite temperature”, Physica A: Statistical Mechanics and its Applications **158**, 1–21 (1989).

[6] L. C. Albuquerque, C. Farina, and S. J. Rabello, “Schwinger’s formula and the partition function for the bosonic and fermionic harmonic oscillator”, (1994).

[7] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2015).

[8] W. P. Su, J. R. Schrieffer, and A. J. Heeger, “Soliton excitations in polyacetylene”, Physical Review B **28**, 2099–2111 (1983).

[9] J. K. Asbóth, L. Oroszlány, and A. Pályi, “The su-schrieffer-heeger (ssh) model”, A Short Course on Topological Insulators Lecture Notes in Physics, 1–20 (2016).

[10] J. P. Hague and C. Maccormick, “Quantum simulation of electron–phonon interactions in strongly deformable materials”, New Journal of Physics **14**, 033019 (2012).

[11] F. L. J. Vos, D. P. Aalberts, and W. V. Saarloos, “Su-schrieffer-heeger model applied to chains of finite length”, Physical Review B **53**, 14922–14928 (1996).

[12] K. A. Chao and Y. Wang, “Lattice vibrations in trans -polyacetylene”, Physica Scripta **34**, 177 (1986).

[13] E. J. Meier, F. A. An, and B. Gadway, “Observation of the topological soliton state in the su–schrieffer–heeger model”, Nature Communications **7** (2016).

[14] H. Takayama, Y. R. Lin-Liu, and K. Maki, “Continuum model for solitons in polyacetylene”, Phys. Rev. B **21**, 2388–2393 (1980).

[15] D. K. Campbell, “Conducting polymers and relativistic field theories”, Synthetic Metals **125**, Celebrating the year 2000 Nobel prize in Chemistry for the discovery and development of conducting polymers, 117–128 (2001).

[16] Y. Kobayashi and H. Minakata, “Thermodynamics of the Su-Schrieffer-Heeger model”, (1994).

[17] H. Caldas, “An effective field theory model for one-dimensional ch chains: effects at finite chemical potential, temperature and external zeeman magnetic field”, Journal of Statistical Mechanics: Theory and Experiment **2011**, P10005 (2011).

[18] G. Profeta, M. Calandra, and F. Mauri, “Phonon-mediated superconductivity in graphene by lithium deposition”, Nature Physics **8** (2012).

[19] B. M. Ludbrook, G. Levy, P. Nigge, M. Zonno, M. Schneider, D. J. Dvorak, C. N. Veenstra, S. Zhdanovich, D. Wong, P. Dosanjh, C. Straßer, A. Stöhr, S. Forti, C. R. Ast, U. Starke, and A. Damascelli, “Evidence for superconductivity in li-decorated monolayer graphene”, Proceedings of the National Academy of Sciences **112**, 11795–11799 (2015).

[20] G. L. Goodvin, “Ground state properties and soliton excitations in polyacetylene”, (2005).

[21] S. Raeisi, N. Wiebe, and B. C. Sanders, “Quantum-circuit design for efficient simulations of many-body quantum dynamics”, New Journal of Physics **14**, 103017 (2012).

[22] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”, Nature **549**, 242–246 (2017).

[23] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. Chan, "Low-depth quantum simulation of materials", Physical Review X (2018).

[24] P. O'Malley, R. Babbush, I. Kivlichan, J. Romero, J. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, E. Jeffrey, A. Megrant, J. Mutus, C. Neil, C. Quintana, D. Sank, T. White, J. Wenner, A. Vainsencher, P. Coveney, P. Love, H. Neven, A. Aspuru-Guzik, and J. Martinis, "Scalable quantum simulation of molecular energies", Physical Review X **6**, 031007 (2016).

[25] J. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms", New Journal of Physics **18**, 023023 (2016).

[26] R. Babbush, D. Berry, I. Kivlichan, A. Wei, P. Love, and A. Aspuru-Guzik, "Exponentially more precise quantum simulation of fermions in second quantization", New Journal of Physics **18**, 033032 (2016).

[27] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. Chan, "Low-depth quantum simulation of materials", Physical Review X **8**, 011044 (2018).

[28] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, "Cloud Quantum Computing of an Atomic Nucleus", (2018).

[29] S. P. Jordan, K. S. M. Lee, and J. Preskill, "Quantum algorithms for quantum field theories", Science **336**, 1130–1133 (2012).

[30] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, "Quantum-Classical Computations of Schwinger Model Dynamics using Quantum Computers", (2018).