IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

Quantum Computing in Estimating Ground State Energies of Systems

Batu Yalcin

Robert College, Arnavutkoy, Kurucesme Cd. No:87 Besiktas, Istanbul, 34345, Turkey; batu.yalcin@gmail.com

ABSTRACT: Knowing the ground state energies of molecules and atoms gives scientists critical insight into these particles' behavior in reactions, thermodynamic and molecular properties, and excitation energies. However, due to the quantum mechanical nature of particles, finding the ground state of a system is intractable, both mathematically and experimentally. Computational methods are utilized to estimate the ground state energies. As we approach the limits of classical computers, quantum computers are viewed as promising alternatives that present a bright future for computational chemistry. In this paper, I review classical and quantum algorithms/methods for computational chemistry. Then, I discuss the current topics in using quantum computers for computational chemistry by analyzing the performance of the Variational Quantum Eigensolver (VQE) in different contexts on IBM's noisy quantum computer simulator, "QasmSimulator," and one of IBM's quantum computers, ibmq_belem. This review presents a theoretical and practical view of quantum computational chemistry. It allows the reader to learn the theory behind the algorithms and implement them with the guidance of the presented code and its detailed explanations. The purpose of this paper is to leave the reader with the necessary theory and adequate programming reference to implement the algorithms in various contexts.

KEYWORDS: Chemistry; Computational Chemistry; Quantum Computing; IBM Qiskit; Variational Quantum Eigensolver.

Glossary

Ground State: The lowest energy state of a system

N-body Problem: The problem of solving the equations of motion for a system consisting of N bodies

State vector: A vector describing the state of a quantum system

Hamiltonian: The energy functional of a system

Wave function: The function that is the solution to the Schrödinger Equation for a system

Ansatz: Initial estimate for a mathematical problem that is fine-tuned and optimized later

TISE: Time-Independent Schrödinger Equation
BOA: Born-Oppenheimer Approximation
HF: Hartree-Fock
SCF: Self-Consistent Field
CI: Configuration Interaction
CC: Coupled-Cluster
CCD: Coupled-Cluster Doubles
CCSD: Coupled-Cluster Singles and Doubles
BD: Brueckner Doubles
UCC(SD): Unitary Coupled-Cluster (Singles and Doubles)
FTQC: Fault-Tolerant Quantum Computing
NISQ: Noisy Intermediate Scale Quantum
HQC: Hybrid Quantum-Classical
QPE: Quantum Phase Estimation
QFT: Quantum Fourier Transform
VQE: Variational Quantum Eigensolver
HEA: Hardware-Efficient Ansätze
PMA: Physically-Motivated Ansätze

Introduction

Particles on atomic and sub-atomic levels, such as electrons and protons, possess a **quantum mechanical** nature; thus, restrictions exist on what can be measured and learned about them. One problem of interest is the ground state energies of molecules, i.e., the lowest energy levels, which are essential for studying molecules and their behavior in chemical reactions. This energy may be expensive to measure in a lab. Furthermore, equations for finding the ground state energy are difficult to solve. This is due to the **Quantum Mechanical Many-Body Problem**, which arises from the difficulty of solving the equations of motion for many interacting bodies. This problem will be discussed further in the next section.

"The underlying physical laws necessary for the mathematical theory of ... the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble." - Paul A. M. Dirac

Due to these limitations, the field of computational chemistry has arisen. The field makes use of computers to approximate valid solutions for the ground state energy and other systems' properties, having many applications such as efficient drug simulation for drug discovery and increased efficiency for batteries, reducing waste, and maximizing power. However, the limitations of classical computers restrict the efficiency of these applications. At this point, quantum computers are promising to provide drastic scaling in terms of computational cost and a significant increase in efficient applications. This paper reviews the application of computational chemistry to estimating ground state energies of systems, starting from classical

DOI: 10.36838/v6i5.5

ods and ending with implementing a quantum algorithm on an actual quantum computer and a simulator. The review presents major topics in quantum chemistry simulations from a semi-empirical standpoint, with a theoretical review and original experiments, which leads to a more comprehensive understanding of the topic. Moreover, for readers who would like to implement what they've read, the final section can work as a benchmark for the code and experiments.

The Quantum Mechanical Many-Body Problem:

To completely describe the quantum mechanical behavior of the electrons in a system, solving for the many-electron wave function of the system via the **Schrödinger Equation** is necessary.[1]

There are two reasons why the problem is intractable:

• There are $N \approx 10^{23}$ electrons in a mole of solid, and there are $3N$ degrees of freedom (i.e., 3D coordinates of each electron).

• The electrostatic forces between electrons (Coulomb Interactions) cause a correlation between the electronic motions; thus, the problem cannot be simplified into N single-body problems rather than an N-body wave function.[2]

The Time-Independent Schrödinger Equation and the Hamiltonian:

The Schrödinger Equation is a linear partial differential equation in terms of the wave function, which predicts the future behavior of a system. The most general form predicts the evolution of a state over time:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$$

where, $|\psi(t)\rangle$ is the state vector of the system, and H is the quantum mechanical Hamiltonian, the operator that gives the energy of the system.

To find the energy levels of a stationary state, an eigenvalue equation called the Time-Independent Schrödinger Equation (TISE) can be derived from the previous equation:

$$H|\psi(x)\rangle = E|\psi(x)\rangle$$

where $\psi(\mathbf{x})$ is the spatial wave function, and E is the energy level of the system. From now on, ψ will indicate the spatial wave function, $\psi(\mathbf{x})$, unless indicated otherwise. To solve for the eigenvalue E, the expectation of the Hamiltonian is taken:[3]

$$\langle H\rangle = \langle\psi|E|\psi\rangle = E\langle\psi||\psi\rangle = E$$

The molecular Hamiltonian, H, is the sum of the potential and kinetic energies of a system:

$$H = T_N + T_e + V_{ee} + V_{eN} + V_{NN}$$

where e denotes electrons, N denotes nuclei, T denotes kinetic energies, and V denotes potential energies.

The Born-Oppenheimer Approximation (BOA) states that because the masses of nuclei are three orders of magnitude greater than the masses of electrons, the wave function ψ and the Hamiltonian H can be separated to yield the electronic wave function, ψ_e and the electronic Hamiltonian, H, which only depend parametrically on nuclear positions, i.e., the nuclear positions are constants that alter the value of the function rather than variables. Using the BOA, the TISE takes a new form:

$$H_e\psi_e = E_e\psi_e$$

in which only the electronic degrees of freedom are needed to be dealt with. Notice that this equation gives the electronic energy, E_e and not the total energy, E. However, E can be obtained by following a similar procedure with the nuclear components of the wave function and the Hamiltonian.[4] Because this problem deals with the electronic structure of the system, it is named the "electronic structure problem."

Although the Born-Oppenheimer approximation helps reduce the complexity of the eigenvalue equation, exponentially large dimensions remain as the wave function cannot be split into N single-body problems. To treat this problem, many approximation techniques are applied to the wavefunction along with the BOA. In the following two sections, these methods will be examined.

Classical Computational Chemistry and Its Limitations:

There are several classical theories to use when dealing with the electronic structure problem: two of the most significant theories are Density Functional Theory (DFT), which removes all the degrees of freedom except electron density, and Molecular Orbital Theory. For the sake of brevity, this section will only encompass Molecular Orbital Theory methods. Within these methods, different methods of approximation will be covered from simple to complex, i.e., Hartree-Fock, Configuration Interaction, and Coupled Cluster methods, depending on the many-body interactions considered in the calculation. However, to tackle a chemistry problem computationally, the first step is to find a way to represent the wave function compactly. Here, Slater Determinants are used.

Slater Determinants:

Slater Determinants, named after John Slater, are a way to represent a wave function in terms of spin orbitals while obeying the "Antisymmetry Principle".

Antisymmetry Principle:

The electronic wave function is antisymmetric under the exchange of the coordinates of two electrons, which can be shown with the equation below for a two-electron wave function:

$$\psi(\mathbf{x_1}, \mathbf{x_2}) = -\psi(\mathbf{x_2}, \mathbf{x_1})$$

Let's construct a product wave function for a two-electron system in terms of spin-orbital functions, χ, which are single electron wave functions taking both the position and spin angular momentum of a particle as its parameters. As an example, let's use the ground state of He, $1s^2\ 2s^0\ 2p^0$. Both electrons occupy the spatial orbital $1s$. However, they occupy different spin orbitals, obeying the **Pauli Exclusion Principle**. Then, the wave function becomes:

$$\psi(\mathbf{x_1}, \mathbf{x_2}) = \chi_1(\mathbf{x_1})\chi_2(\mathbf{x_2})$$

After the permutation of electrons, i.e., the interchange of the spin orbitals each electron occupies, the wave function becomes:

$$\psi(\mathbf{x_2}, \mathbf{x_1}) = \chi_1(\mathbf{x_2})\chi_2(\mathbf{x_1})$$

This product wave function does not satisfy the indistinguishability principle and results in two totally different functions. To prevent this, the two-electron wave function can be written as **a linear combination of functions** that satisfy the normalization condition. Since each electron is equally likely to occupy each spin-orbital, an expression for the two-electron wave function in terms of spin orbitals can be found:

$$\psi(\mathbf{x_1}, \mathbf{x_2}) = \tfrac{1}{\sqrt{2}}[\,\chi_1(\mathbf{x_1})\chi_2(\mathbf{x_2}) - \chi_1(\mathbf{x_2})\chi_2(\mathbf{x_1})]$$

The above equation satisfies the Antisymmetry Principle and the normalization condition, which are respectively shown below:

$$\psi(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{\sqrt{2}}[\,\chi_1(\mathbf{x_1})\chi_2(\mathbf{x_2}) - \chi_1(\mathbf{x_2})\chi_2(\mathbf{x_1})]$$

$$= -\frac{1}{\sqrt{2}}[\,\chi_1(\mathbf{x_2})\chi_2(\mathbf{x_1}) - \chi_1(\mathbf{x_1})\chi_2(\mathbf{x_2})] = -\psi(\mathbf{x_2}, \mathbf{x_1})$$

$$\left(\frac{1}{\sqrt{2}}\right)^2 + \left(-\frac{1}{\sqrt{2}}\right)^2 = 1$$

This equation also satisfies the Pauli Exclusion Principle, i.e., two electrons cannot occupy the same spin orbitals, as when $\mathbf{x}_1=\mathbf{x}_2$, $\psi=0$.[5]

When the number of possible spin orbitals increases, this notation is impractical to use. However, determinants can be used to express the wave functions. The expression found is equivalent to a **Slater Determinant** for two spin orbitals:

$$\psi(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{\sqrt{2}}\begin{vmatrix}\chi_1(\mathbf{x_1}) & \chi_2(\mathbf{x_1})\\ \chi_1(\mathbf{x_2}) & \chi_2(\mathbf{x_2})\end{vmatrix}$$

Generalizing the Slater Determinant to an N-electron wave function, the equation becomes:

$$\psi(\mathbf{x}) = \psi(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}) = \frac{1}{\sqrt{N!}}\begin{vmatrix}\chi_1(\mathbf{x_1}) & \chi_2(\mathbf{x_1}) & \ldots & \chi_N(\mathbf{x_1})\\ \chi_1(\mathbf{x_2}) & \chi_2(\mathbf{x_2}) & \ldots & \chi_N(\mathbf{x_2})\\ \vdots & \vdots & \ddots & \vdots\\ \chi_1(\mathbf{x_N}) & \chi_2(\mathbf{x_N}) & \ldots & \chi_N(\mathbf{x_N})\end{vmatrix}$$

To introduce a briefer notation, ψ could be expressed in ket notation including only the occupied orbitals, i.e., $\psi=|\chi_i\,\chi_j \ldots \chi_k\rangle=|ij\ldots k\rangle$.[6]

Self-Consistent Field (SCF)/Hartree-Fock Method:

The Hartree-Fock (HF) method, which assumes that the wave function can be expressed as a single Slater Determinant, is equivalent to the assumption that the movement of each electron is independent of other electrons except the Coulomb repulsion caused by the mean field, which depends on the average position of all electrons. As each electron is evaluated self-consistently, this method is called the Self-Consistent Field (SCF) method.

Under this assumption, the **variational theorem**, which is a theorem that states that the trial energy, i.e., the energy given by the trial wave function, cannot be lower than the actual ground state energy, can be used. In the HF method, this trial wave function is a single Slater Determinant. The Hartree-Fock method aims to find the best possible single electron wave functions, χ, functions that will minimize the total energy given by:

$$E = \langle\psi_{\mathrm{HF}}|H_e|\psi_{\mathrm{HF}}\rangle$$

By using the variational method, the parameters of ψ can be adjusted by solving for the Hartree-Fock equations, which are eigenvalue equations that give the energy and Hartree-Fock Orbital for one electron. These minimizations will result in the "best single Slater Determinant." The limit of the lowest energy that can be obtained by a wave function described by a single Slater Determinant is called the **Hartree-Fock limit**, the energy is called the **Hartree-Fock Energy**, the best possible single-electron orbitals are called the **Hartree-Fock Orbitals**, and the Slater Determinant formed by these orbitals is called the **Hartree-Fock wave function**.[7]

Although the assumption that a wave function can be represented in terms of a single Slater Determinant may result in poor computational results for many systems, the HF wave function can be used as a starting point to construct improved wave function ansätze, i.e., the initial estimate for a mathematical problem that is fine-tuned and optimized later.[8]

Configuration Interaction:

In a quantum system, the motions of electrons cannot be described only with their independent behavior and the Coulomb repulsion force that is exerted on them, as the behavior also depends on the correlation between the electrons caused by their location. However, approximating the wave function by a single Slater Determinant does not consider the spatial correlations in the electron motion. This correlation can be accounted for by expressing the wave function as a linear combination of different electronic configurations, represented by Slater Determinants. The electronic combinations are basis functions, which are all possible configurations of electrons in an N-electron system represented by Slater Determinants.[9] The linear combination of these Slater Determinants regards the interaction of different configurations; thus, this method is called **Configuration Interaction (CI)**. Mathematically, an N-electron wave function can be represented as a linear combination of a complete set of N-particle basis functions, which are constructed from a complete set of 1 particle basis functions.[10] Denoting the ith wave function represented by a single Slater determinant as Φ_i, and the number of basis functions with d, the wave function is equal to:

$$\psi_{\mathrm{CI}} = \sum_{i=1}^{d} a_i \Phi_i$$

where the coefficients a_i are variationally determined. Solving for this equation gives an exact solution for the wave function.[8,11] Then, the total energy is given by:

$$E = \langle\psi_{\mathrm{CI}}|H_e|\psi_{\mathrm{CI}}\rangle$$

In real life, a complete set of 1-particle basis functions cannot be found because there are infinitely many 1-electron wave functions. Therefore, an incomplete set of 1-electron wave functions is used, which is assumed to be enough. In theory, calculating the wave function with an incomplete set of 1-electron basis functions but all possible N-electron basis functions is called a "full CI." If the set of 1-electron basis functions is

DOI: 10.36838/v6i5.5

complete, which never is the case in practice, the method is called "complete CI."

However, full CI can also not be carried out on a classical computer because all the cap *N*-particle basis wave functions "can not be efficiently stored and manipulated."[8] Therefore, different CI methods beyond this work's scope are used to approximate the wave function.

Coupled Cluster:

Nowadays, Coupled Cluster (CC) methods are the preferred methods for high accuracy in computations. Coupled Cluster methods were found to be used in nuclear physics, and it took a while to include electron correlation in wave function estimation. In CC theory, the trial wave function, usually the Hartree-Fock wave function, ψ_0 is operated on by an exponential function $e^{\hat{T}}$, where:

$$\psi_{CC} = e^{T}\psi_0$$

and the total energy is given by:

$$E = \langle\psi_{CC}|H_e|\psi_{CC}\rangle$$

Note that CC methods use an exponential operator while CI methods express the wave function as a linear combination. The cluster operator $\hat{T}$ is defined as the sum of excitation operators $\hat{T}_i$:

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \ldots + \hat{T}_N$$

Each excitation operator $\hat{T}_i$ generates all ith excited electron configurations that are represented in terms of a Slater Determinant:

$$\hat{T}_1\psi_0 = \sum_{i}^{occ}\sum_{a}^{virt} t_i^a\,\psi_i^a$$

$$\hat{T}_2\psi_0 = \sum_{i<j}^{occ}\sum_{a<b}^{virt} t_{ij}^{ab}\,\psi_{ij}^{ab}$$

$$\hat{T}_k\psi_0 = \sum_{i_1<i_2<\ldots<i_k}^{occ}\ \sum_{a_1<a_2<\ldots<a_k}^{virt} t_{i_1 i_2 \ldots i_k}^{a_1 a_2 \ldots a_k}\,\psi_{i_1 i_2 \ldots i_k}^{a_1 a_2 \ldots a_k}$$

where "occ" denotes the occupied molecular orbitals, "virt" denotes virtual (unoccupied) spin orbitals, t are coefficients and ψ are Slater Determinants of different configurations, as when an electron is excited, it leaves an occupied orbital and occupies an unoccupied orbital.

The exponential of the operator $\hat{T}$ can be expressed by using the McLaurin expansion of e^x:

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots = \sum_{n=0}^{\infty}\frac{x^n}{n!}$$

Substituting $\hat{T}$ for x:

$$e^{\hat{T}} = 1 + \hat{T} + \frac{\hat{T}^2}{2} + \frac{\hat{T}^3}{6} + \ldots = \sum_{n=0}^{\infty}\frac{\hat{T}^n}{n!}$$

To deal with a more straightforward equation to compute, some restrictions are made on $\hat{T}$. For example, in Coupled Cluster Doubles (CCD), $\hat{T}$ is approximated by $\hat{T} \approx \hat{T}_2$.

However, limiting the cluster operator, $\hat{T}$ does not restrict the excitations to a given level. This can be seen by considering the fact that $\hat{T}_2^2$ is two double excitation operators combined, which is a quartet excitation operator. There also are different restrictions, such as Coupled Cluster Singles and Doubles (CCSD), that make the approximation:[12]

$$\hat{T} \approx \hat{T}_1 + \hat{T}_2$$

CCSD is exact for two-electron systems due to the fact that, at most, double-excited Slater Determinants can be generated for a two-electron system. In a two-electron system, only the first few terms of $e^{\hat{T}}$ gives non-zero values when operated on by the Hamiltonian, as it only contains one and two electron operators. Thus, the infinite series can be truncated into a finite sum with the first few terms.

"CCSD is the only pure CC method that can be used in routine applications," as higher term approximations become more expensive. However, some variations can be made on the CCSD method, e.g., CCSD (T), where triples can also be estimated with some trade-offs. Bruckner's (B) theory is a variation of the CC theory. The Brueckner Doubles (BD) method can be compared to CCSD in terms of both accuracy and computational cost.[12,13]

A difference in CC theory from the aforementioned algorithms is that the energy obtained is non-variational, meaning it can also be less than the true energy of the ground state. This problem is solved with Variationally Optimized Couple Cluster; however, it has significant limitations on a classical computer.

In CC theory, the cluster operator may not necessarily be a pure excitation operator. For example, in Unitary Coupled Cluster, the cluster operator is an operator that has the excitation and de-excitation operator combined in one unitary operator $(U = e^{\hat{T}-\hat{T}^{\dagger}})$.

Although introducing the de-excitation operator has its advantages, the number of terms in the expectation value $\langle\psi_{UCC}|He^{\hat{T}}|\psi_{UCC}\rangle$, become infinite and the problem becomes intractable on classical computers. However, on a quantum computer, the operations on qubits, the quantum analogs of bits, are generally realized in terms of unitary operations, which makes preparing a UCC ansatz, i.e., trial wave function, efficient if the cluster operator can be represented as a low-depth quantum circuit.[8]

The realization of the UCC ansatz is one of the many advantages quantum computers bring to the world of computational chemistry. In the next section, these advantages will be explored.

Quantum Computation for Computational Chemistry:

"Nature is not classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it does not look so easy." - Richard Feynman

This is a quote taken from the 1981 lecture of Feynman, proposing the first idea of a quantum computer.[14] Due to the fact that quantum computers leverage two significant

properties in quantum mechanics, i.e., superposition and entanglement, they reduce the computational cost of important problems in computation. For example, the time it takes to factorize an integer, the key idea that lies underneath most encryption systems (RSA Encryption), can be drastically reduced using a quantum algorithm called Shor's algorithm. However, these algorithms cannot be implemented yet, as the capacity to complete a task of any size without error or with minimal error has not reached an adequate level.

In this section, quantum computation will be introduced first. Then, different computational chemistry algorithms on quantum computers, categorized according to their hardware requirements, will be presented and discussed.

Introduction to Quantum Computation:

The smallest unit of information in a quantum computer is a qubit, the quantum analog of a bit, which takes the values 1 or 0 in a classical computer. The 0 state on a qubit is denoted by $|0\rangle$ and the 1 state is denoted by $|1\rangle$, which is called the bra-ket notation where $\langle|$ is the bra, and $|\rangle$ is the ket. Different from a bit, these qubits can represent a linear combination of states $|0\rangle$ and $|1\rangle$, which is called a superposition. A superposition is only preserved until it is observed (or measured). When observed, the state collapses into $|0\rangle$ or $|1\rangle$. A qubit in the same superposition state may give a different outcome when measured, as there is a probability for the qubit to collapse to both states.

In general, the state of a qubit is in the form:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle, |c_0|^2 + |c_1|^2 = 1$$

for c_0, $c_1 \in C$, where $|c_0|^2$ is the probability of the qubit collapsing into $|0\rangle$, and $|c_1|^2$ is the probability of the qubit collapsing into $|1\rangle$.

A well-known superposition state is:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Where $|\psi\rangle$ denotes the qubit's state. Note that for this specific state, the magnitudes of the coefficients are equal, meaning the probability of measuring $|0\rangle$ is equal to the probability of measuring $|1\rangle$, which is $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$.
The qubit having an equal
probability of collapsing into both states is what makes this state, which is denoted by $|+\rangle$, important.

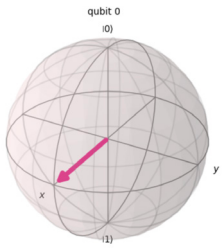

Figure 1*: Bloch Sphere representing the state $|+\rangle$. Notice that the two poles represent $|0\rangle$ and $|1\rangle$, and the vector representing $|+\rangle$ is equidistant from the vectors representing $|0\rangle$ and $|1\rangle$.

The state of a qubit is represented on a Bloch Sphere. A Bloch Sphere representing $|+\rangle$ can be found in Figure 1.

When a qubit is observed, its superposition collapses into either the $|0\rangle$ state or the $|1\rangle$ state, with probabilities $|c_0|^2$ and $|c_1|^2$, respectively. Quantum computers leverage the interference property as a quantum circuit interferes with the components of the superposition.[15,16]

Quantum circuits are meaningful when they act on multiple qubits, e.g., the encryption method that is most used today, 2048-bit RSA encryption, can be cracked with ≈4000 ideal qubits.[17]

Let's look at how multi-qubit systems can be represented and how entanglement can be leveraged in these systems. A system with two qubits in the state $|0\rangle$ can be represented as $|00\rangle$ where the leftmost number in the ket is the state of the first qubit. The entanglement of multiple qubits has important applications in quantum computing. For example, entangling the second qubit with the first qubit in a two-qubit system while the first qubit is in the superposition $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
will yield the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$$

On a circuit diagram, this circuit (with measurement at the end) is shown as in Figure 2:

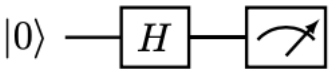

Figure 2: Single qubit first set to an equal superposition, then measured.

The circuit above is the circuit describing a qubit initialized to $|0\rangle$, a Hadamard gate applied to it followed with a measurement. At the end, with the measurement gate, the probability to measure each state is 1/2. The $|\Phi^+\rangle$ state,

$$\frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$$

is drawn as in Figure 3:

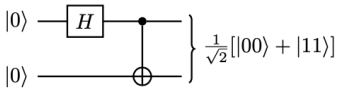

Figure 3: Quantum Circuit for a Bell State

where the two-qubit gate after the H gate is the CNOT gate, which entangles two qubits, the filled dot denoting the control qubit and the dot with a plus in it denoting the target qubit. If the control qubit is in the state $|0\rangle$, nothing changes in the target qubit; however, if the control qubit is in the state $|1\rangle$, the state of the control qubit is reversed.

Let's work through preparing the $|\Phi^+\rangle$. When the control qubit is $|0\rangle$, the target qubit will not change; it will stay $|0\rangle$ forming the state $|00\rangle$; however, when the control qubit is $|1\rangle$, the target qubit will be reversed, which means it will become $|1\rangle$ forming the state $|11\rangle$. Since it is equally likely for the control qubit to collapse into both states,
the resulting state is $\frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$.

****:** All figures in this text either were ensured to be licensed appropriately or are original work.

DOI: 10.36838/v6i5.5

This subsection has explored the properties that make quantum computers advantageous. By leveraging these properties, namely preparing states that cannot be prepared classically, superposition, entanglement, and interference, quantum computers can provide accurate results for many problems in chemistry, such as finding the ground state energy of quantum systems.

Algorithms for Fault-Tolerant Quantum Computing (FTQC):

Quantum computers are highly promising devices in terms of solving complex chemistry problems. However, these promises have yet to be realized due to the physical limitations of current technologies for building quantum computers. Physical qubits, as of right now, cannot work precisely as an ideal qubit for several reasons, e.g., the *noise* caused by physical conditions such as "control electronics, heat, or impurities in the qubit material,"[19] the decay of a qubit from state $|1\rangle$ to $|0\rangle$ (relaxation time), and the losing of quantum coherence of qubits (*dephasing time*).[20]

Due to these physical limitations, it takes multiple physical qubits to realize an ideal qubit. These promising algorithms need fault-tolerant quantum computing (FTQC) to work, which will take millions of physical qubits to realize. Nevertheless, theoretical algorithms have been constructed for these FTQCs to revolutionize the field of computation when they are built.

Hamiltonian Evolution:

One problem that can be tackled using FTQCs is evolving the Hamiltonian over time. Although finding zero-temperature ground states from the TISE is useful, many chemical processes include time evolution and take place at finite temperatures. Therefore, solving for the Time-Dependent Schrödinger Equation is quite helpful in describing many processes. The time-dependent Schrödinger Equation (when the reduced Planck's constant $\hbar$ is omitted):

$$i\frac{d}{dt}\psi(\vec{r},t) = H\psi(\vec{r},t)$$

which can be formally solved as follows when the Hamiltonian is time-independent:

$$\psi(\vec{r},t) = e^{-iHt}\psi(\vec{r},0)$$

Expanding this equation in terms of the eigenvectors ψ_j of the time-independent Hamiltonian:

$$\psi(\vec{r},t) = \sum_j c_j\, e^{-iE_jt}\psi_j(\vec{r})$$

From this equation, the difficulty of this method becomes apparent. To simulate the Hamiltonian over time exactly, the full spectrum of the Hamiltonian eigenvectors and the expansion of the initial state must be known. This is intractable on classical computers for systems except for systems of very small sizes.[8] To deal with this, several approximations that are beyond the scope of this work are made, e.g., molecular dynamics.

Several approaches, not discussed in detail, are present to deal with this simulation problem on quantum computers, e.g., product formula algorithms, a linear combination of unitaries, quantum walks, quantum signal processing, and quantization.

Quantum Phase Estimation:

Quantum Phase Estimation (QPE) is an algorithm that estimates the phase φ in binary representation when a unitary operator $U=e^{i\varphi}$, i.e. an operator that is equal to the identity operator when multiplied by its conjugate transpose ($UU\dagger=I$), and an approximation of its corresponding eigenstates are given.

In the case of finding ground state energies, the unitary operator used is $U=e^{-iHt}$), which simulates the Hamiltonian evolution over time. To implement this operator in terms of quantum gates, H is first written as a product of non-commuting terms, $H = \sum_{j=1}^{m} H_j$. Then, a Trotter-Suzuki approximation is made, i.e., breaking down the Hamiltonian time evolution operator, e^{-iHt}, as a product of time evolution operators, e^{-iHjt}, that are easy to simulate.[21] Afterwards, the QPE algorithm is used to extract information about the eigenvalue spectrum of the Hamiltonian.

The scheme of the Quantum Phase Estimation algorithm is as follows: First, all ancilla (auxiliary) qubits are put in an equal superposition state via Hadamard gates. Then, the Hamiltonian time evolution operator e^{-iHt} is applied as several controlled unitaries from the ancilla register to the register that is initialized in an eigenstate of the Hamiltonian. Thus, the eigenvalue corresponding to the eigenstate is printed on the ancilla register as phases via a phenomenon called phase kickback. Finally, using Inverse Quantum Fourier Transform, these eigenenergies are extracted from the phases and are measured.[22]

In the QPE, the phase imprinted on the kth qubit is $e^{-2^k i\lambda_m t}$ where λ_m is the eigenvalue corresponding to ψ_m. Imprinting this state is shown below:

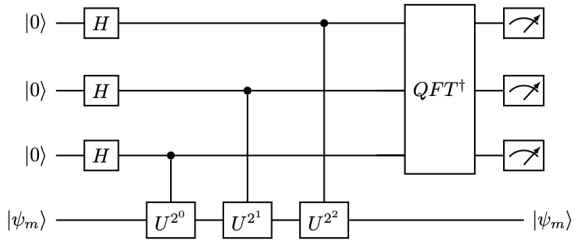

Figure 4: QPE with 3 Ancilla Qubits

where QFT† is the inverse of Quantum Fourier Transform, U^{2^k} is the unitary operator applied 2^k times to the qubit, and $|\psi_m\rangle$ is the trial wave function.

First, the phase kick-back trick is used. In this trick, a phase is added to one of the states in the superposition by using multi-qubit operators. In this case, the unitary operator U^{2^k} adds the $e^{2^k i\lambda_m t}$ phase to the $|1\rangle$ state.

However, an exact eigenstate of the Hamiltonian, ψ_m cannot be always prepared. The second register may be a superposition of the eigenstates, expressed as:

and the phase estimation will yield:

$$\sum_m a_m |\psi_m\rangle$$

where $\widetilde{\lambda_m}$ is an approximation of the eigenvalue λ_m.

$$\sum_m a_m |\tilde{\lambda}_m t/2\pi\rangle|\psi_m\rangle$$

Then, the probability of the second register to collapse to eigenstate λ_m is $|a_m|^2$. This shows the importance of state preparation, as the initial state should be dominated by the overlap with the ground state to obtain the ground state eigenenergy.

Then, the inverse Quantum Fourier Transform is applied to the qubits, which transforms the state into a computational basis state $|x_1\, x_2 \ldots x_N\rangle$ where $\lambda_m t/2\pi \approx 0.x_1\, x_2 \ldots x_N$[8]. It has been shown that with success probability p, the number of precise bits that can be calculated, ω is:[23]

How the inverse Quantum Fourier Transform works will be examined in the next section.

$$\omega = N - \lceil \log_2 \left(2 + \frac{1}{2p}\right) \rceil$$

Quantum Fourier Transform:

The Quantum Fourier Transform (QFT) is the quantum analog of the Discrete Fourier Transform. The QFT maps a state on a computational basis to a state on a Fourier basis, with a similar formula to its classical version. The circuit diagram of QFT with four qubits can be drawn as in Figure 5:

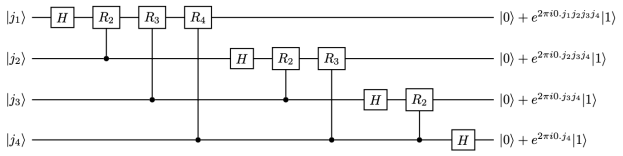

Figure 5: QFT with 4 Qubits

where R_k acts as the following operation on the two-qubit state $|x_c x_t\rangle$:

$$R_k|0x_t\rangle = |0x_t\rangle, R_k|1x_t\rangle = e^{\frac{2\pi i}{2^k}x_t}|1x_t\rangle$$

The Quantum Fourier Transform can simply be expressed as:

$$\mathrm{QFT}|x\rangle = |\tilde{x}\rangle$$

Where the state in the computational basis is $|x\rangle$ and the state in the Fourier basis is $\tilde{x}$.The formula is as follows for $N=2^n$ in a quantum system with n qubits:[8]

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/2^n} |k\rangle$$

When k is written in binary form $k=k_1\, k_2 \ldots k_n$, $k/2^n = \sum_{j=1}^{n} k_j/2^j$

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/2^n} |k\rangle$$

Representing this equation as a product of n-qubits that make up the system, the equation becomes:[24]

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2}x}|1\rangle\right)\left(|0\rangle + e^{\frac{2\pi i}{2^2}x}|1\rangle\right) \ldots \left(|0\rangle + e^{\frac{2\pi i}{2^{n-1}}x}|1\rangle\right)\left(|0\rangle + e^{\frac{2\pi i}{2^n}x}|1\rangle\right)$$

Transforming $|x\rangle$ into binary representation $|x_1\, x_2 \ldots x_n\rangle$,

$$\mathrm{QFT}|x_1 x_2 \ldots x_n\rangle = \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i 0.x_n}|1\rangle)(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle) \ldots (|0\rangle + e^{2\pi i 0.x_1 x_2 \ldots x_{n-1}x_n}|1\rangle)$$

Notice that the kth factor in this equation is in a similar form with the result of the unitary applied to the kth qubit in the phase kickback phenomenon, i.e. $\left(|0\rangle + e^{-2^k i \lambda_m t}|1\rangle\right)$,

when λ_m is represented in

binary form. From this similarity, the use of inverse Quantum Fourier Transform, $\mathrm{QFT}^\dagger$ to extract the eigenenergies can be understood intuitively.

Although these algorithms are seemingly promising, it is not possible to implement these algorithms meaningfully on quantum computers of today. FTQCs are to come; however, the devices used now are noisy, and intermediate scale. In the next section, algorithms for today's quantum computers will be examined.

Hybrid Quantum-Classical Algorithms for the Noisy Intermediate-Scale Quantum Era (NISQ):

Today's quantum computers are quite limited compared to FTQCs. They are noisy, and they do not accommodate quantum error correction; thus, algorithms must have low circuit depth and must be done within the decoherence time. The results should also be accessible by simple measurements. These devices are called Noisy Intermediate-Scale Quantum (NISQ) devices. However, they can still surpass classical computers on suitable and relevant problems.

The limitations of NISQ computers have motivated Hybrid Quantum-Classical (HQC) algorithms to be developed, in which the algorithm benefits from the advantages of both devices. The general working scheme of HQC algorithms is as follows: First, some parametrized circuit is passed to the quantum computer. Then, through measurements, the desired quantity, e.g., the expectation value of a Hamiltonian, is computed and passed to the classical computer. According to some updating algorithms and the computed value, new parameters are calculated and passed back to the quantum circuit. This loop continues until a threshold to obtain the desired result is reached.[8]

In the context of computational chemistry, the Variational Quantum Eigensolver (VQE) has been one of the first proposed HQC algorithms.[8] In the next section, the VQE will be explained in detail.

Variational Quantum Eigensolver (VQE):

The main idea of the Variational Quantum Eigensolver (VQE) is the *Variational Principle*.[25] Algorithms obeying the variational principle are guaranteed to provide upper bounds for the true ground state energy (λ_{gs}):

$$\langle \psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle = \lambda_\theta \geq \lambda_{gs}$$

Thus, the parametrized state $\psi(\theta)$ can be optimized to return the minimum value of λ_θ.

The VQE algorithm consists of four steps:

1. *Mapping to a Hamiltonian:* Map the ground state problem to a Hamiltonian represented as a sum of Pauli strings. Pauli strings consist of consecutive operations with Pauli matrices, which are easily implementable gates on a quantum computer.

where h_i are coefficients and P_i are Pauli strings.

$$H = \sum_i h_i P_i$$

DOI: 10.36838/v6i5.5

2. *State Preparation:* A trial wave function, which is parametrized ($|\psi(\theta)\rangle$) is prepared on the quantum computer. This wave function can be chosen according to the properties it should display, for example the Hartree-Fock wave function might be used as a trial. Then, a unitary operator that is also parametrized ($U(\theta)$), which is generally hard to implement classically, is applied to the trial wave function. The unitary depends on the choice of ansatz. In choosing the ansatz, two main approaches are seen:

a) Hardware-Efficient Ansätze (HEA): HEA makes use of the hardware of quantum computers to prepare states by sequences of single and two-qubit operations on the computer. The reason these ansätze are called hardware efficient is that they consider the connectivity specifications of the hardware, building a circuit that is realizable with the specific hardware. However, making the circuit hardware efficient requires exploring a bigger portion of the Hilbert Space, resulting in a higher circuit depth.[26] The high number of operations require error mitigation, making the algorithm more noise-prone.

b) Physically Motivated Ansätze (PMA): Physically Motivated Ansätze are techniques based on exact wave function approximation techniques. A widely used example of PMA is the Unitary Coupled Cluster, which is an ansatz that uses the unitary operator $e^{\hat{T}-\hat{T}^\dagger}$ approximates the wave function,

where $\hat{T}$ is the cluster operator which was explained in detail. Although it is intractable to simulate the operator classically, unitary operators such as this one are easy to simulate on quantum computers. However, as the ansatz is "agnostic" to the hardware,[26] the circuit depth needed to implement the ansatz on today's limited hardware is large. Thus, this ansatz may pose a problem for NISQ devices. After the proposal of UCC, new and improved PMA such as Qubit Coupled Cluster (Coupled Cluster method built-in directly in the qubit space), variations on UCCSD (Unitary Coupled Cluster Singles Doubles), and ADAPT-VQE (VQE in which the operators that are used are selected from a pool in each step, growing the ansatz to optimize the algorithm) have been proposed.[27]

State preparation is important, as the importance of correct trial wave function choice has been highlighted in several research. It has been proven that there are areas where the cost function of the algorithm, i.e., an indicator of how good the algorithm is, does not change much, causing the optimization process to lengthen. These areas are referred to as "barren plateaus."[26] In these areas, the gradient of the cost function exponentially vanishes. The choice of ansatz is essential to the convergence of the cost function and avoiding the "barren plateaus."

Estimate the energy of the Hamiltonian by measuring the expectation of the individual Pauli strings ($\langle P_i \rangle$) and sum these terms up accordingly to estimate the energy of the Hamiltonian due to the linearity of the expected value function. This is called *Hamiltonian averaging*:

$$H = \sum_i h_i P_i$$

3. Update the parameters, θ classically, and repeat until the energy converges. The optimization algorithms in this stage are still subject to research, as there are a lot of limitations, such as the effects of noise on the parameter space.[8,28]

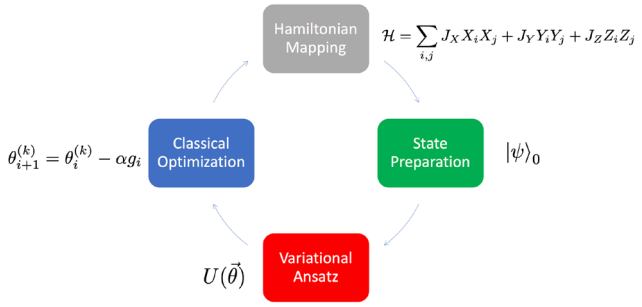

Figure 6: Overview of VQE

An illustration of these steps can be found in Figure 6.

VQE is a promising area of research with a broad spectrum of application areas. For example, the company IonQ has partnered with Hyundai Motors to produce more efficient Lithium batteries, and they are using VQE for the simulation of the batteries.[29]

The following section will show an implementation of VQE on a Lithium hydride molecule.

Implementing a Quantum Algorithm: Estimating the Ground State Energy of LiH:

This section will explore the implementation of VQE using different optimizers and ansätze. With this exploration, VQE will be covered more deeply, touching on important limitations such as the "barren plateau" problem and differences between ansätze. Finally, VQE will be implemented on a real quantum computer of IBM, ibmq_belem.

Methods and Technical Details

Several implementations of VQE with varying ansätze and optimizers were explored. Properties of different ansätze and optimizers can be found below:

• Ansätze:

- UCCSD: Unitary Coupled Cluster Singles and Doubles (UCCSD) is a PMA in which a unitary operator $U = e^{\hat{T}-\hat{T}^\dagger}$ used, where $\hat{T}$ is a cluster operator. When the Singles and Doubles variation is used, the cluster operator, $\hat{T}$ is approximated as $\hat{T} \approx \hat{T}_1 + \hat{T}_2$.

However, the hardware of NISQ devices has important limitations, which pose problems in efficiently building a circuit that accommodates the ansatz.

- Hardware-Efficient SU2 2-local circuit: Hardware-efficient circuits leverage the quantum mechanical nature of the quantum hardware. SU2 means "special unitary group of degree 2," which is a group that contains unitary matrices that have the dimensions 2×2, and have their determinant equal to 1.[30] Pauli rotation gates, e.g., Pauli X, are some members of the SU2 group. The 2-local part indicates that two-qubit gates can be used at maximum. This means the circuit will only contain two-qubit CNOT gates and single-qubit rotation gates to

operate on the wave function. The increased circuit depth in these ansätze makes them more noise-prone.[26]

• Optimizers

- COBYLA: The Constrained Optimization by Linear Approximation (COBYLA) is a linear optimization algorithm that does not use the cost function's gradient (gradient-free). In COBYLA, the actual optimization problem is approximated with a linear programming problem in each iteration. After each iteration, the approximate problem is improved to better estimate the actual problem.[31]

- SLSQP: The Sequential Least Squares Programming (SLSQP) is a sequential programming algorithm, i.e., there is a deterministic sequence of steps in the algorithm that optimize the cost function according to the gradient.[32,33]

- SPSA: The Simultaneous Perturbation Stochastic Approximation (SPSA) is a stochastic optimization algorithm. The advantage of SPSA lies in the fact that with every step, the parameters are changed in a "properly" random fashion, hence stochastic, at the same time, therefore simultaneous perturbation, which results in only two measurements of the cost function needed regardless of the parameter count. SPSA is designed to accommodate noisy measurements;[34] thus, it proves useful in NISQ devices.

Different implementations of VQE and their purposes in the research are outlined below:

• An Efficient SU(2) ansatz was implemented using the three classical optimizers on two simulators, one noisy and one ideal, to observe their susceptibility to noise.

• The convergences of these optimizers were graphed out to see the "barren plateau" problem explicitly. The barren plateau problem is the problem where the gradient of the cost function vanishes exponentially at an undesired point (the desired point is the global minimum in this case). The number of steps needed for convergence drastically increases as the algorithm falsely decides on the global minimum at another point. The system size and noise can cause the barren plateau problem.

• An Efficient SU(2) and a UCCSD ansatz were implemented on a noisy simulator using a COBYLA optimizer to observe the susceptibility of the ansätze to noise on a connection-wise unlimited hardware.

• An Efficient SU(2) and a UCCSD ansatz were implemented on the ibmq_belem computer using a COBYLA optimizer to observe the susceptibility of the ansätze to noise on a connection-wise limited hardware.

• An Efficient SU(2) ansatz with COBYLA as the classical optimizer was implemented on a noisy simulator. An informed initial point was used for the ansatz in one optimization, while a random initial point was used for the other. The purpose was to see the difference setting an informed initial point makes, as setting a poor initial point is also a driver of barren plateaus.[26]

Code Overview:

The VQE algorithm was implemented using Qiskit (0.37.0), Qiskit Nature (0.4.2) and Qiskit IBM Runtime (0.5.0). For mathematical operations and array manipulation, numpy was utilized while matplotlib.pyplot was used for graphing. The base code for the optimizers was inspired by the Qiskit Textbook[35] and Documentation.[36] The necessary packages are imported below:

```
from qiskit_nature.drivers import Molecule
from qiskit_nature.drivers.second_quantization import ElectronicStructureDriverType,ElectronicStructure
MoleculeDriver
from qiskit_nature.mappers.second_quantization import ParityMapper
from qiskit_nature.converters.second_quantization import QubitConverter
from qiskit_nature.problems.second_quantization import ElectronicStructureProblem
from qiskit_nature.transformers.second_quantization.electronic import ActiveSpaceTransformer
from qiskit.algorithms import NumPyMinimumEigensolver
from qiskit_nature.algorithms import GroundStateEigensolver, NumPyMinimumEigensolverFactory
from qiskit.algorithms.optimizers import SLSQP, COBYLA, SPSA
from qiskit.providers.aer import QasmSimulator, StatevectorSimulator
from qiskit.circuit.library import EfficientSU2, TwoLocal, SXGate
from qiskit_nature.circuit.library import UCCSD
from qiskit.algorithms import VQE
import matplotlib.pyplot as plt
import numpy as np
from time import time
```

Then, the initialization function for the LiH molecule at different interatomic distances is defined. The PySCF driver is used to interface Qiskit with the corresponding computational chemistry program.

```
distance_array = np.arange(0.5, 4.0, 0.1) #List for different interatomic distances in the LiH molecule

def initializeMolecule(distance): #initializing the molecule with a specific interatomic distance
    moleculeLiH = Molecule(
        geometry=[["Li", [0.0, 0.0, 0.0]], ["H", [0.0, 0.0, distance]]], charge=0, multiplicity=1
    )
    driverLiH = ElectronicStructureMoleculeDriver(
        moleculeLiH, basis="sto3g", driver_type=ElectronicStructureDriverType.PYSCF #PYSCF Interface F
or HF Solutions
    )
    return driverLiH
```

After the initialization, the function to convert the molecule into an ElectronicStructureProblem, a "problem" object that is "solvable" with Qiskit, should be defined. During the conversion, the molecule is transformed using the active space transform, and the explored space is limited to its active space. The active space includes the orbitals that are/can be partially occupied (active orbitals). For the LiH molecule, as 2 of the 4 electrons stay in the 1s orbital of Li, not contributing to the bonding, that orbital (core orbital) is disregarded, along with the 2 electrons, leaving the other two. Three molecular orbitals are chosen to consider, as more than three would result in >5 qubits, and less would result in reduced accuracy. As the accessible quantum computer for this research had five qubits, this transform was chosen. The molecular orbital diagram of LiH can be found in Figure 7:

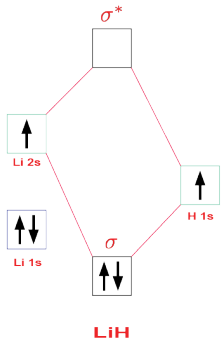

Figure 7: MO Diagram of LiH[37]

```
active_space_transformer = ActiveSpaceTransformer(
    num_electrons=2,
    num_molecular_orbitals=3,
)
def problemize(driver):
    es_problem = ElectronicStructureProblem(driver, [active_space_transformer])
    return es_problem
```

The Hamiltonian should be mapped onto qubits to tackle the problem on a quantum computer. To reduce the number

DOI: 10.36838/v6i5.5

of qubits needed by exploiting the symmetries in the Hamiltonian (2 qubit reduction), the parity mapper is used to define the converter.

```
qubit_converter_parity = QubitConverter(mapper=ParityMapper(), two_qubit_reduction=True)
#exploit symmetries for a reduction by two qubits
```

The qubit operator and the electronic structure problem are defined for each interatomic difference defined before (0.5, 0.6, ..., 3.9).

```
op_list = []
es_list = []

for i in distance_array:
    driver = initializeMolecule(i)
    es_problem = problemize(driver)
    sqops_es = es_problem.second_q_ops()
    hamiltonian = sqops_es[0]
    qubit_op = qubit_converter_parity.convert(hamiltonian, num_particles=es_problem.num_particles)
    op_list.append(qubit_op)
    es_list.append(es_problem)
```

Then, the exact ground state energies should be found as reference values. The energies found with the code are not the full ground state energies but the reduced electronic energy. The energy extracted with the active space transform and nuclear repulsion energies can be found later and are easier to compute.

```
np_solver = NumPyMinimumEigensolver()
np_results = []
np_energies = []
for i in range(len(distance_array)):
    np_result = np_solver.compute_minimum_eigenvalue(op_list[i])
    np_results.append(np_result)
    np_energies.append(np_result.eigenvalue)
    print("Interatomic Distance:", distance_array[i], "Exact Energy:", np_energies[i])
```

Now, the noisy and ideal simulators, i.e., QasmSimulator and StatevectorSimulator, are used to simulate the Efficient SU(2) ansatz with COBYLA, SLSQP, and SPSA optimizers. First, the ansatz should be defined. By default, three repetitions of the rotation and entanglement block are used, resulting in 32 parameters. The ansatz is drawn after initialization. As it is seen in Figure 8, the ansatz consists of single qubit Ry, Rz rotation gates and CNOT two-qubit entanglement gates:

```
HEansatz = EfficientSU2(num_qubits = 4, entanglement = "linear")
HEansatz.decompose().draw('mpl', style='iqx')
```

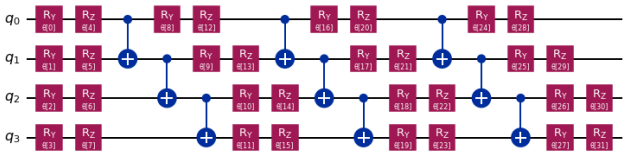

Figure 8: 4 Qubit Hardware-Efficient SU2 2-Local Circuit Ansatz

There are 32 parameters, $\theta_0,\theta_1,\ldots,\theta_{31}$, that will be classically optimized. An initial point is defined as one whose dimensions are equal to the number of parameters. For better results, this initial point can be determined from prior knowledge and defined as a closer point to the optimal point.

```
#Defining A Random Initial State
num_parameters = HEansatz.num_parameters
np.random.seed(24) #reproducibility
init = np.random.random(num_parameters)
```

After that, the optimizers are defined as follows:

```
#Defining the Optimizers to be Used
optimizers = [SPSA(maxiter = 500), COBYLA(maxiter = 500), SLSQP(maxiter = 500)]
```

Then, a callback function is defined to monitor convergence and the arrays to store the intermediate values.

```
#Defining A Callback Function To Monitor Convergence
evals = []
energies = []
def vqe_callback(num_evals, parameters, energy, stdev):
    evals.append(num_evals)
    energies.append(energy)
#Defining Arrays to Store Different Callback Results (wrt Optimizers)
callback_cnts = [np.empty(len(distance_array), dtype = object)]*len(optimizers)
callback_engs = [np.empty(len(distance_array), dtype = object)]*len(optimizers)
```

Then, the energies are computed:

```
HE_energies=[[],[],[]]
times = []
t1 = time()
for j in range(len(optimizers)):
    vqe = VQE(ansatz = HEansatz, optimizer = optimizers[j], quantum_instance = QasmSimulator(), initial_
point = init)
    print("Using",optimizers[j])
    for i in range(len(distance_array)):
        print(i)
        result = vqe.compute_minimum_eigenvalue(op_list[i])
        HE_energies[j].append(np.real(result.eigenvalue))
        print("Interatomic Distance:", distance_array[i], "Estimated Energy:", result.eigenvalue, "Exact Energ
y:", np_energies[i])
    t2 = time()
    times.append(t2-t1)
```

Finally, the results are graphed.

```
#Plot the comparison with exact energies
plt.plot(distance_array,HE_energies[0], color = "blue", label = "HE(SPSA)")
plt.plot(distance_array,HE_energies[1], color = "green", label = "HE(COBYLA)")
plt.plot(distance_array,HE_energies[2], color = "black", label = "HE(SLSQP)")

plt.plot(distance_array,np_energies, color = "red", label = "Exact energies")

plt.xlabel("Interatomic Distance")
plt.ylabel("Ground State Energy")
plt.legend()
plt.show()
```

Then, the convergence of each optimizer was monitored when computed using an arbitrary interatomic distance, 2.4 Angstrom. The convergence was graphed.

```
optimizer1 = SLSQP(maxiter= 500)
vqe_mconv1 = VQE(ansatz = HEansatz, optimizer = optimizer1, callback = vqe_callback, quantum_insta
nce = QasmSimulator(), initial_point = init)
evals = []
energies = [] #resetting
result1 = vqe_mconv1.compute_minimum_eigenvalue(op_list[19]) #19 is just an arbitrary choice

SLSQP_cnts = np.asarray(evals)
SLSQP_engs = np.asarray(energies)

plt.plot(SLSQP_cnts, SLSQP_engs, label = "SLSQP")
plt.axhline(y=np_energies[19], color='r', linestyle='--',label="Exact")
plt.xlabel("Evaluation Count")
plt.ylabel("Energy")
plt.title("VQE convergence for SLSQP")
plt.legend()
plt.show()
```

The remaining implementations of VQE can be achieved similarly, and the corresponding code will not be included for brevity purposes.

Results and Discussion

Noise Susceptibility of Optimizers:

A Hardware-Efficient ansatz was used to compare the noise propensities of the optimizers COBYLA, SLSQP, and SPSA (Unless noted otherwise, the HEA with a linear connectivity scheme in Figure 9 will be used afterward).

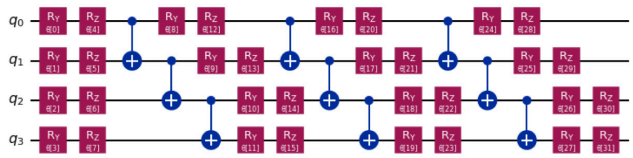

Figure 9: HE Ansatz with a Linear Connectivity Scheme

Each optimizer was given a maximum iteration limit of 500. The performance of each optimizer can be found in Figure 10:

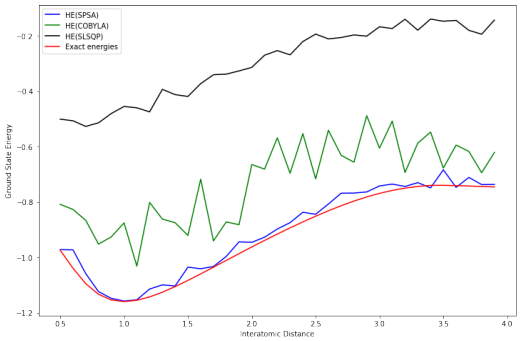

Figure 10: Performance of Optimizers According to Interatomic Distance. SPSA resulted in improved converges compared to SPSA, COBYLA, SLSQP.

SLSQP has performed more poorly than the other optimizers, while SPSA has performed the best. As is seen in Figure 11, although SLSQP converged steadily in the ideal simulator, it was not able to converge on the noisy simulator. This can be mostly attributed to a noise-induced barren plateau, as the optimizer decided that the value has converged with an error percentage of ≈72.6% and stopped iterating even before the 35th iteration. Figure 12 shows a similar but less drastic error has happened using COBYLA. In the ideal simulator, COBYLA has not decided on convergence, but it was cut off by the number of maximum iterations allowed. In the noisy simulator, COBYLA did not even reach 250 iterations and decided on a value with an error percentage of ≈41.3%. SPSA, an optimizer designed to accommodate noise, performed the best in the noisy simulator. From Figure 13, although not fully converging in the ideal simulator, due to the limit on maximum number of iterations, SPSA has performed better than the other optimizers on the noisy simulator, being cut off by the maximum number of iterations at a value with an error percentage of ≈5.5%. SPSA proved to be the most noise-prone optimizer, while SLSQP proved to be the least.

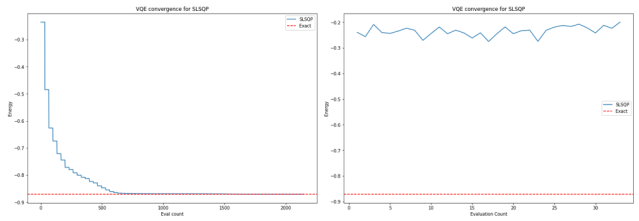

Figure 11: Ideal vs Noisy SLSQP

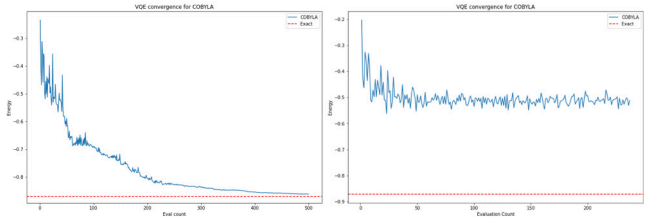

Figure 12: Ideal vs Noisy COBYLA

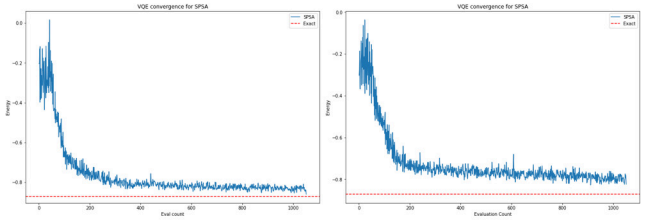

Figure 13: Ideal vs Noisy SPSA

Results: Noise affected the convergence of SPSA significantly less than COBYLA and SLSQP.

Importance of Setting an Initial Point:

In the previous subsection, all the VQE algorithms started from a randomly chosen initial point. However, it is known that a poor selection of the initial point can also induce barren plateaus.[26] This initial point can be the HF solution, which is easier to find, etc. This subsection will explore the performance improvement in COBYLA when an informed initial point is set. The initial point used is found from running the algorithm with SPSA, which has 250 as the maximum number of iterations.

As seen from Figure 14, when an informed initial point was given, the error percent was reduced from ≈27% to nearly ≈3%. An important thing to notice is that this error percentage is less than the error percentage of SPSA performing on a noisy simulator. Moreover, the total optimization time of finding the initial point with the SPSA and finding the optimal point with COBYLA is drastically less than full optimization with SPSA, with the former being around 75 seconds and the latter being around 100 seconds. Using each optimizer to alleviate their strengths can be a favorable choice. Here, SPSA was used to evade noise-induced barren plateaus early in the optimization, and COBYLA was used for fast convergence to finish the optimization.

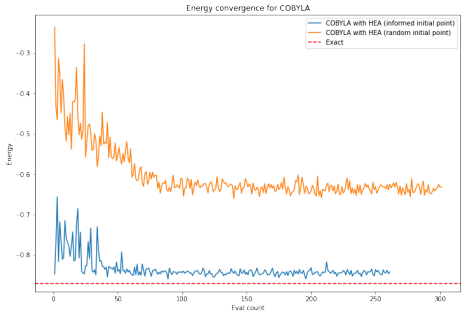

Figure 14: Informed vs Random Initial Point with COBYLA Optimizer

Results: An informed initial point significantly improved convergence for COBYLA.

Observing the Difference Between Ansätze:

Theoretically, physically-motivated ansätze should experience less noise and need less circuit depth than hardware-efficient ansätze. However, in practice, the connectivity limitations of computers can make a difference, as the number of CNOT gates needed to implement an arbitrary entanglement scheme can increase due to the connectivity limitations. To observe this difference, a VQE with COBYLA as the optimizer was run with an Efficient SU2 ansatz, which accommodates the limitations of ibmq_belem, and a

DOI: 10.36838/v6i5.5

hardware-agnostic UCCSD ansatz. The ansätze can be seen in Figure 15:

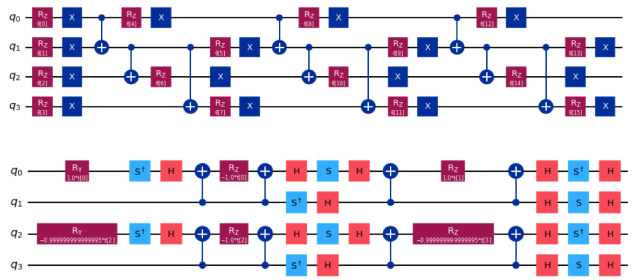

Figure 15: Hardware-Efficient vs Physically-Motivated Ansatz(excerpt)

More specifically, the former ansatz has its rotation gates as the RZ and X gates (the SX gate was not included for the sake of simplicity) and the entanglement gate as the CNOT gate. Also, the entanglement scheme obeyed the "pyramidal" connectivity scheme of ibmq_belem, as seen in Figure 16. The UCCSD ansatz did not follow this scheme. For example, the entanglement between q2 and q3 would not be possible on ibmq_belem.

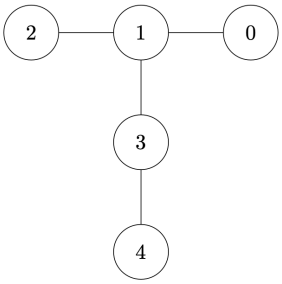

Figure 16: Connectivity Scheme of ibmq_belem

On the QasmSimulator, the UCCSD ansatz performed a lot better than the HEA, as expected, because the ansatz has a chemical intuition behind it and needs to span less of the Hilbert Space to find the optimal solution. The convergence graph can be seen in Figure 17. On the simulator, the error percentage for the HEA was ≈19%, while it was ≈3% for the UCCSD ansatz. When the connectivity limitations came in and the algorithm was run on ibmq_belem, the error percentage for the HEA rose to ≈* 31%. In comparison, the error percentage for the UCCSD ansatz rose to ≈61%, becoming nearly twice of HEA's.

Moreover, the time UCCSD took was ≈94 seconds more of the hardware-efficient ansatz's, the former being ≈1190 seconds and the latter being ≈1094 seconds. Thus, with limited connectivity, UCCSD ansätze tends to perform more poorly, causing the need for Hardware-Efficient Ansätze. An important thing to note here is that the accuracy is relatively low in both ansätze, which can be attributed to the far-from-ideal hardware used in this experiment. More error mitigation and better connectivity would be essential for useful computations.

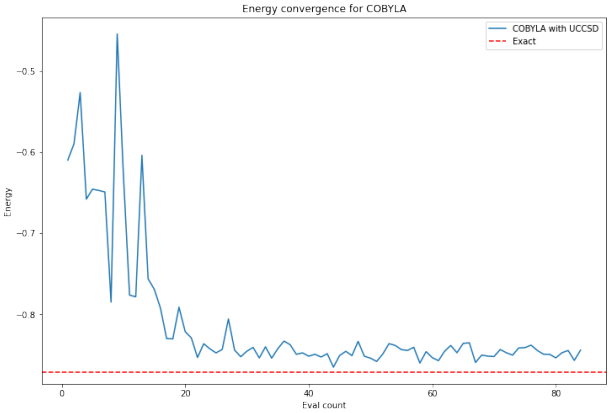

Figure 17: Convergence of UCCSD Ansatz with COBYLA

Summary of Results

Both components of VQE, i.e., classical optimization and quantum computational ansätze, are ongoing topics for research. The experiment that was discussed demonstrated how different ansätze and classical optimizers affect the results of the VQE. This research has demonstrated several aspects of barren plateaus, such as its various causes, i.e., noise and poor initial point selection, and the effect of different ansätze and optimizers on their occurrences. SPSA, a noise-prone optimization algorithm, was able to overcome noise-induced barren plateaus. On the contrary, SLSQP and COBYLA experienced barren plateaus. Now, it is valid to ask why COBYLA, a gradient-free algorithm, experiences barren plateaus if barren plateaus occur with the exponential vanishing of the gradient. It has been found that, for gradient-free optimizers such as COBYLA that work according to the difference in two points of the cost function, barren plateaus are still a problem since the difference between points also vanishes exponentially.[39] Thus, COBYLA was also subject to barren plateaus along with SLSQP.

Along with this, SPSA took more time for optimization, and combining different optimizers/using optimizers with conscious initial points proved to provide faster and more accurate computation. It was shown that in a noisy but fully connected hardware, UCCSD ansatz greatly outperformed the Efficient SU2 ansatz. Still, in a connectivity-wise limited hardware, the Efficient SU2 ansatz outperformed the UCCSD ansatz. It was demonstrated that hardware limitations, such as noise, connectivity, etc., are important obstacles in front of the quantum revolution in computational chemistry and, more generally, quantum computing. Ongoing research on both sides, the hardware and the software (e.g., ansätze and optimizers), can result in more accurate and faster chemistry calculations in the future.

Quantum Algorithms Versus Algorithms on Classical Computers:

It is seen that solving for the ground state of the Lithium hydride molecule numerically is better than the VQE algorithm in terms of accuracy, and the time needed is relatively low. Now, a question may arise: Why is VQE used? Although the empirical results may be deceptive in terms of the performance of VQE, it should be kept in mind that the advantage of quantum computers arises in more complex systems. The

DOI: 10.36838/v6i5.5

* This specific excerpt of the ansatz can be accommodated by using the last four qubits instead of the first four, but the whole ansatz includes other connections that make it impossible to directly implement on ibmq_belem

memory and time complexity of methods on classical computers grow exponentially with the system size, whereas the same qualities grow polynomially in quantum computers. The reason is that an n-qubit system can be used to represent a Hamiltonian with a size exponential to the system size, as the n-qubit system can produce 2^n states; however, this is not possible on classical computers due to the nature of bits.

In the experiment discussed above, a quantum simulator with five qubits is used. Considering that Google was able to demonstrate the quantum advantage with 53 qubits in 2019[38], it is natural that our quantum algorithm ran on a 5-qubit quantum computer did not surpass a classical method. However, the comparison between optimizers and ansätze in the experiment, combined with theoretical knowledge, provides accurate insight into the limitations of different methods while implementing VQE.

Future Directions

As technology develops, lots of new application areas for quantum computational chemistry will occur. Promising areas of employing computational chemistry include drug discovery, simulating proteins and how they bind with each other, simulating the 3D potential energy surface of molecules, increasing energy efficiency through simulating materials such as batteries, etc. Let's dive into how quantum computers will affect the field of drug discovery and energy efficiency.

Quantum Computing in Energy Efficiency:

Automotive industry giants, e.g. Daimler and Hyundai, are partnering with quantum computing companies such as IBM, and IonQ, trying to improve the capacity and performance of batteries for electric vehicles (EVs). For example, on 20th January 2022, Hyundai announced its partnership with IonQ. Peter Chapman, CEO and president of IonQ, explained the rationale behind the partnership with quantum computers: it becomes possible to ensure that one is extracting maximum efficiency and eliminating sources of potential waste. Hyundai is looking to optimize their lithium compound batteries with the assistance of quantum computers, and IonQ's device has recently outperformed all other devices tested in a series of benchmarking tests run by industry consortium QED-C and which is currently in private beta.[31]

Quantum Computing in Drug Discovery:

Protein simulation is essential in discovering new drugs. However, simulating proteins and their folding are hard problems for classical computers due to limitations revolving around tasks such as sampling and simulating possible protein conformations (configuration of amino acids) as the number of possible combinations increases exponentially with the number of amino acids. These limitations can be overcome using quantum computers. In the long term, quantum computers are hypothesized to help form structure-related hypotheses in Computer-Aided Drug Design (CADD). For short-term applications, quantum computers are envisioned to assist CADD by increasing the accuracy of protein models.[40]

Researchers have taken on this task by forming a Hamiltonian, including the superposition of all possible physically meaningful conformations, and sampling the Hamiltonian statistically to find the most stable conformation.[41] Using a modified VQE called CVaR (Conditional Value-at-Risk) VQE, which can be applied to any polymer and not just proteins, they have simulated Angiotensin, a protein that increases blood pressure and is made up of a ten amino-acid chain, with 22 qubits, and a neuropeptide that is made up of a seven amino-acid chain with 20 qubits efficiently.[41-43]

On the other hand, only a small number of amino acid chains can be simulated on NISQ devices. Researchers have proposed an algorithm for finding stable conformations with FTQC, using a well-known purely quantum algorithm called Grover's Search, which provides an exponential speed up to unstructured search problems. The time complexity of the algorithm is $O\left(\frac{N}{M}\right)$.

where N is the number of possible states, in this case, conformations, and M is the number of states that are the answer to the problem. In contrast, the time complexity of a classical algorithm for unstructured search would be $O\left(\sqrt{\frac{N}{M}}\right)$.[44]

However, the proposed algorithm cannot be yet implemented due to the hardware-related limitations of quantum computers.

One of the most prominent future applications of quantum computers is the simulation of substances ranging from batteries to proteins. These simulations can enhance the quality of human life, both industrially and biologically, increasing efficiency in most material-related tasks.

Conclusions

Finding the ground state energy of molecules is crucial in predicting and understanding their behavior. To accomplish this, purely mathematical techniques come short, and computational chemistry is needed. However, as humanity is reaching the limits of classical computers, a considerable list of computational chemistry tasks still remains intractable on a classical computer. To perform these tasks, quantum computers, which significantly reduce the effect of the system size on the task, are starting to be utilized.

Quantum computers are in an early era, the Noisy Intermediate-Scale Quantum (NISQ) Era, where the capabilities of quantum computers are significantly limited compared with the capabilities of Fault-Tolerant Quantum Computing (FTQC). Although this is the case, NISQ-era devices can still outperform classical computers at specific tasks with specific algorithms. An example is using the VQE algorithm for ground state estimation, which was discussed in detail in this paper. Although the algorithm is promising, there are significant obstacles to overcome. Nevertheless, the field of computational chemistry has gained considerable room for improvement with quantum computers. If this opportunity is seized, exciting news is awaiting humanity.

Acknowledgment

First, I would like to thank my supervisor for this project and my chemistry teacher, Hazal Özilhan, for her outstanding support, help, and feedback throughout the project. I would also like to acknowledge QubitxQubit for a tremendous introductory Quantum Computing course, which opened my eyes to this exciting field. Finally, I would like to thank my family,

DOI: 10.36838/v6i5.5

who cultivated my enthusiasm for the field and created a supportive environment.

References

1. 400 Jenkins, Stephen. "3. The Many Body Problem and Density Functional Theory." University of Exeter Physics and Astronomy, U of Exeter, 13 Jan. 1997, newton.ex.ac.uk/research/qsystems/people/jenkins/mbody/mbody3.html. Accessed 28 Feb. 2022.
2. "The Many-Body Problem." Condensed Matter Theory, Durham University, 16 Dec. 2004, cmt.dur.ac.uk/sjc/thesis_prt/node22.html. Accessed 28 Feb. 2022.
3. Shankar, Ramamurti. "The Postulates –A General Discussion." Principles of Quantum Mechanics, 2nd ed., e-book ed., New York City, Plenum Press, 1994. pdf.
4. Feiguin, Adrian E. "The Born-Oppenheimer approximation." Northeastern University, 4 Nov. 2009, web.northeastern.edu/afeiguin/phys5870/phys5870/node9.html. Accessed 28 Feb. 2022.
5. Antisymmetric Wavefunctions Can Be Represented by Slater Determinants. 19 Mar. 2020, https://chem.libretexts.org/@go/page/210844.
6. —. "Slater Determinants." The Sherrill Group, The Sherrill Group Georgia Institute of Technology, 30 May 2002, vergil.chemistry.gatech.edu/notes/hf-intro/node4.html. Accessed 28 Feb. 2022.
7. Hanson, David M., Erica Harvey, Robert Sweeney, and Theresa J. Zielinski. The Self-Consistent Field Approximation (Hartree-Fock Method). Chemical Education Digital Library (ChemEd DL), 23 July 2021, https://chem.libretexts.org/@go/page/4537.
8. Cao, Yudong, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. "Quantum Chemistry in the Age of Quantum Computing." Chemical Reviews, vol. 119, no. 19, 30 Aug. 2019, pp. 10856-915. ACS Publications, https://doi.org/10.1021/acs.chemrev.8b00803.
9. Hanson, David M., Erica Harvey, Robert Sweeney, and Theresa J. Zielinski. Configuration Interaction. Chemical Education Digital Library (ChemEd DL), 23 July 2021, https://chem.libretexts.org/@go/page/4531.
10. Sherrill, David. "An Introduction to Configuration Interaction Theory." Chemistry and Biochemistry, Georgia Institute of Technology, 1995. The Sherrill Group, vergil.chemistry.gatech.edu/notes/ci.pdf. Accessed 28 Feb. 2022.
11. Simons, Jack. "Configuration State Functions Can Express the Full N-Electron Wavefunction." University of Utah, 10 Aug. 202, https://chem.libretexts.org/@go/page/63313.
12. Hofmann, Matthias, and Henry F. Schaefer. "Computational Chemistry." Encyclopedia of Physical Science and Technology, 2003, pp. 487-506, https://doi.org/10.1016/B0-12-227410-5/00129-0.
13. Zhang, Igor Ying, and Andreas Grüneis. "Coupled Cluster Theory in Materials Science." Frontiers in Materials, vol. 6, 5 June 2019, https://doi.org/10.3389/fmats.2019.00123. Accessed 28 Feb. 2022.
14. Preskill, John. "Quantum Computing 40 Years Later." Feynman Lectures on Computation. arxiv.org, https://doi.org/10.48550/arXiv.2106.10522.
15. "Single Qubit Gates." Qiskit, 13 July 2021, qiskit.org/textbook/ch-states/single-qubit-gates.html. Accessed 28 Feb. 2022.
16. "Quantum computing in a nutshell." Qiskit, qiskit.org/documentation/qc_intro.html. Accessed 28 Feb. 2022.
17. Baumhof, Andreas. "Breaking RSA Encryption – an Update on the State-of-the-Art." Quintessence Labs, 13 June 2019, www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art/. Accessed 28 Feb. 2022.
18. Albash, Tameem, and Daniel A. Lidar. "Adiabatic Quantum Computation." Reviews of Modern Physics, vol. 90, no. 1, 29 Jan. 2018. arxiv.org, https://doi.org/10.1103/RevModPhys.90.015002.
19. Matheson, Rob. "Uncovering the hidden 'noise' that can kill qubits." MIT News, 16 Sept. 2019, news.mit.edu/2019/non-gaussian-noise-detect-qubits-0916. Accessed 28 Feb. 2022.
20. Youssef, Rahaf. Measuring and Simulating T1 and T2 for Qubits. United States: N. p., 2020. Web. doi:10.2172/1656632.
21. "Simulating Hamiltonian Dynamics." Microsoft Technical Documentation, Microsoft, 29 Jan. 2022, docs.microsoft.com/en-us/azure/quantum/user-guide/libraries/chemistry/concepts/algorithms. Accessed 28 Feb. 2022.
22. Whitfield, James D., Jacob Biamonte, and Alán Aspuru-Guzik. "Simulation of Electronic Structure Hamiltonians Using Quantum Computers." Molecular Physics, vol. 109, no. 5, 10 Mar. 2011, pp. 735-50. arxiv.org, https://doi.org/10.1080/00268976.2011.552441.
23. Mcardle, Sam, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum Computational Chemistry." Reviews of Modern Physics, vol. 92, no. 1, 30 Mar. 2020. arxiv.org, https://doi.org/10.1103/RevModPhys.92.015003.
24. "Quantum Fourier Transform." Qiskit, qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html. Accessed 28 Feb.2022.
25. "Simulating Molecules Using VQE." Qiskit, qiskit.org/textbook/ch-applications/vqe-molecules.html. Accessed 28 Feb. 2022.
26. Tilly, Jules, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. 'The Variational Quantum Eigensolver: a review of methods and best practices.' 2021. Web.
27. Grimsley, H.R., Sophia E. Economou, Edwin Barnes, and Nicolas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. Nat Commun 10, 3007 (2019). https://doi.org/10.1038/s41467-019-10988-2.
28. Fedorov, Dmitry A., Bo Peng, Niranjan Govind, and Yuri Alexeev. "VQE Method: A Short Survey and Recent Developments."Materials Theory, vol. 6, no. 1, 6 Jan. 2022, doi:10.1186/ s41313-021-00032-6.
29. Choi, Charles Q. "Hyundai partners with IonQ to optimize lithium-air batteries." IEEE Spectrum, 20 Jan. 2022, spectrum.ieee.org/lithium-air-battery-quantum-computing. Accessed 28 Feb. 2022.
30. "EfficientSU2." Qiskit, qiskit.org/documentation/stubs/qiskit.circuit.library.EfficientSU2.html. Accessed 28 Feb. 2022.
31. https://handwiki.org/wiki/COBYLA Magee, Jeff.
32. "Sequential Programming." Imperial College London, https://www.doc.ic.ac.uk/ jnm/concurrency/online/concurrent/sld007.htm. Accessed 28 Feb. 2022.
33. "NLopt Algorithms." NLopt Documentation, nlopt.readthedocs.io/en/latest/NLopt_Algorithms. Accessed 28 Feb. 2022.
34. Spall, James C. "Simultaneous Perturbation Stochastic Approximation." Johns Hopkins Applied Physics Laboratory, Johns Hopkins University, 29 Mar. 2001, www.jhuapl.edu/spsa/. Accessed 28 Feb. 2022.
35. "Simulating Molecules using VQE." Qiskit, qiskit.org/textbook/ch-applications/vqe-molecules.html. Accessed 14 July 2022.
36. "Qiskit 0.37.0 Documentation." Qiskit, qiskit.org/documentation/. Accessed 14 July 2022.
37. Molecular Orbital Diagram of Heteronuclear Diatomic Molecule LiH. Chemistry LibreTexts, 22 July 2021, chem.libretexts.org/Courses/Lafayette_College/CHEM_212_213%3A_Inorganic_Chemistry_(Nataro)/02%3A_Molecules/2.05%3A_Molecular_Orbitals/2.5.05 3A_Molecular_Orbital_Diagrams. Accessed 10

Apr. 2022.
38. Arute, Frank, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habe gger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Mar kus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Jul ian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fed or Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmi try Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McE wen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud M ohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Nei ll, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. P latt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nich olas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyansk iy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Ben jamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. "Quantum sup remacy using a programmable superconducting processor." Natur e, vol. 574, no. 7779, 23 Oct. 2019, pp. 505–510, https://doi.org/1 0.1038/s41586-019-1666-5.
39. Arrasmith, Andrew, M. Cerezo, Piotr Czarnik, Lukasz Cincio, P atrick J. Coles. "Effect of barren plateaus on gradient-free optimi zation." Quantum, vol. 5. arxiv, arxiv.org/abs/2011.12245. Access ed 16 July 2022.
40. Evers, Matthias, Anna Heid, Ivan Ostojic. "Pharma's digital Rx: Quantum computing in drug research and development." McKin sey & Company, 18 June 2021, www.mckinsey.com/industries/lif e-sciences/our-insights/pharmas-digital-rx-quantum-computing -in-drug-research-and-development. Accessed 24 Apr. 2022.
41. Robert, A., Panagiotis Kl. Barkoutsos, Stefan Woerner, and Ivan o Tavernelli. Resource-efficient quantum algorithm for protein fol ding. npj Quantum Inf 7, 38 (2021). https://doi.org/10.1038/s41 534-021-00368-4
42. Letzter, Rafi. "A Novel Quantum Algorithm for Protein-Foldin g: Paving the Way Toward Resolving One of the Biggest Mysteri es in Biology With Quantum Computers." Medium, 19 Aug. 20 21, medium.com/qiskit/a-novel-quantum-algorithm-for-protein -folding-paving-the-way-toward-resolving-one-of-the-biggest-861112139ff0. Accessed 24 Apr. 2022.
43. Mayo Clinic. 13 Aug. 2021, www.mayoclinic.org/diseases-condi tions/high-blood-pressure/in-depth/angiotensin-ii-receptor-bloc kers/art-20045009. Accessed 24 Apr. 2022.
44. Khatami, Mohammad Hassan, Udson C. Mendes, Nathan Wie be, Philip M. Kim. Gate-Based Quantum Computing for Protein Design. 1, arXiv, 2022, doi:10.48550/ARXIV.2201.12459.
45. Matthew Treinish, Matthew Treinish, Jay Gambetta, Paul Natio n, qiskit-bot, Paul Kassebaum, Diego M. Rodríguez, Salvador de la Puente González, Shaohan Hu, Kevin Krsulich, Jake Lishman, Jim Garrison, Laura Zdanski, Jessie Yu, Manoel Marques, Julien Gacon, Luciano Bello, David McKay, Juan Gomez, Lauren Capel luto, Travis-S-IBM, Ashish Panigrahi, lerongil, Rafey Iqbal Rahm an, Steve Wood, Toshinari Itoko, Christopher J. Wood, Divyansh u Singh, Drew, Eli Arbel, and Glen. Qiskit/qiskit: Qiskit 0.37.0. 0.37.0, Zenodo, 9 Feb. 2022, p., doi:10.5281/zenodo.6784303.

Author

Batu Yalcin is a high school student at Robert College, Istanbul. He's interested in physics, especially spin models, quantum computing, AI, EdTech, and music. He is a National Mathematical Olympiad Silver Medalist and Junior Balkan Mathematics Olympiad Bronze Medalist. Thrilled by the field of quantum computing, he aspires to be a significant contributor in the future.

DOI: 10.36838/v6i5.5