

# VARIATIONAL QUANTUM EIGENSOLVER

Thesis

Submitted in partial fulfilment of the requirements for the degree of

**MASTER OF SCIENCE**

in

**PHYSICS**

by

**VISHNU MK**

**(216PH036)**

Under the guidance of  
**Dr. Deepak Vaid**



PHYSICS DEPARTMENT  
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA  
SURATHKAL, MANGALORE -575025

June, 2022

## **DECLARATION**

I, hereby declare that the report of P.G Project Work entitled “Variational Quantum Eigensolver” which is being submitted to the National Institute of Technology Karnataka, Surathkal, in partial fulfilment of the requirements for the award of the Degree of Master of Science in the Department of Physics, is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

**NITK SURATHKAL**  
**June 2022**

**VISHNU MK**  
**216PH036**

# CERTIFICATE

This is to certify that the P.G. Project Work Report entitled “**Variational Quantum Eigensolver**” submitted by **VISHNU MK, (Roll Number: 216PH036)** as the record of the work carried out by him, is accepted as the **P.G. Project Work Report Submission** in partial fulfilment of the requirements for the award of the degree of **Master of Science** in the **Department of Physics**.

**External Guide**

**Internal Guide**

**Dr.A.V. Narasimhadhan**

(Name and signature with date and seal)

**Dr Deepak Vaid**

(Name and signature with date and seal)

Chairman- DPGC

(Signature with date and seal)

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to my project advisor and project coordinator Dr Deepak Vaid, Department of Physics, NITK, Surathkal for his guidance, instruction and for the valuable project opportunity that he provided for me. Specially, I would like to thank my parents for their support and for having faith in me.

**Place: NITK SURATHKAL**  
**Date: Jan 2022**

**VISHNU MK**  
**216PH036**

## **Abstract**

Quantum Computation techniques have been used to solve the Deuteron problem namely to obtain the eigenstates and eigenvalues of the groundstate of Deuteron and other molecules. Encoding schemes JWT are discussed to convert the Hamiltonian into sum weighted Pauli matrices. Quantum algorithm VQE, is explored to obtain the desired properties. These algorithms are created such that it can be used to any nuclear many-body physics problems.

# Contents

Acknowledgements . . . . .	i
Abstract . . . . .	i
<b>1 Introduction to Quantum Computation</b>	<b>1</b>
<b>2 Introduction to VQE</b>	<b>4</b>
2.1 VQE Overview . . . . .	6
<b>3 Future Scope of VQE</b>	<b>7</b>
<b>4 Variational principle in Quantum Mechanics</b>	<b>9</b>
<b>5 Nuclear physics to Quantum Computation problem</b>	<b>12</b>
5.1 The Nuclear Hamiltonion . . . . .	12
5.1.1 Second Quantization . . . . .	14
5.1.2 Second Quantized fock states . . . . .	14
5.2 Hamiltonion in action . . . . .	15
5.3 Mapping the Hamiltonion to physical Qubits . . . . .	17
5.3.1 Jordan-Wigner Mapping . . . . .	17
<b>6 The Variational Quantum Eigensolver</b>	<b>18</b>
6.1 Variational Ansatz . . . . .	18
6.2 Parameter Optimization . . . . .	19
6.3 Various Quantum Algorithms . . . . .	20
<b>7 VQE Results for Deutron</b>	<b>21</b>
7.1 The Deutron . . . . .	21
7.2 Simulator results . . . . .	21

7.2.1	User defined Ansatz . . . . .	22
7.2.2	QISKIT inbuilt Ansatz . . . . .	22
7.3	Why QASM and statevector simulators giving different results? . . .	23
<b>A</b>	<b>Ground state of Deuteron</b>	<b>25</b>
<b>B</b>	<b>QISKIT version details</b>	<b>31</b>

# Chapter 1

## Introduction to Quantum Computation

The use of quantum mechanics' superposition, interference, and entanglement in computation processes is known as quantum computing. Quantum computers are equipment used for doing quantum calculations. Although current quantum computers are too small to perform better than conventional (classical) computers for practical applications, larger realisations are thought to be capable of solving some computational problems, such as integer factorization (which is the basis of RSA encryption), significantly faster than conventional computers. Quantum information science includes the study of quantum computing as a discipline.

The fundamental units of information in classical computation are bits, which have a state space limited to the binary values "0" and "1." In contrast, the fundamental units of information in quantum computation are qubits, each of which is a two-dimensional Hilbert space, and which, because they are quantum mechanical objects, can remain even in a superposition of the two states. As seen in Figure 1 Figure 7.1, a qubit state can be seen on a Bloch sphere. On the Bloch sphere, a qubit is capable of existing in any state (i.e., at any position on the surface of the sphere).

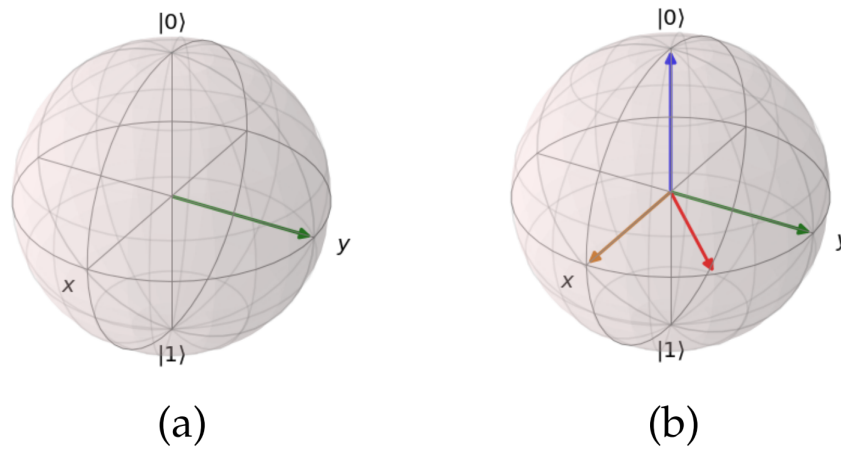
The evolution of bits are governed by logical operators like OR, AND and XOR gates and the evolution of qubits are carried out by unitary operators called gates. These operators preserve norm of the states.



If the choice of basis states are,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1.1)$$

then the state represented Fig. 1(a) can be written as  $|\psi\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\frac{\pi}{2}}|1\rangle)$  then the state represented Fig. 1(a) can be written as  $|\psi\rangle$  a general state can be written as  $|\psi\rangle = a|0\rangle + b|1\rangle$ . If there multiple qubits describing the system then the combined state of the system is written as  $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle \otimes |\psi_4\rangle \otimes \dots$  where  $|\psi_i\rangle$  are individual system states.



**Figure 1.1:** Qubit represented on a Bloch sphere, one qubit(a), multiple qubits(b)

**Some of the gates frequently used are :**

Pauli spin matrices

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (1.2)$$

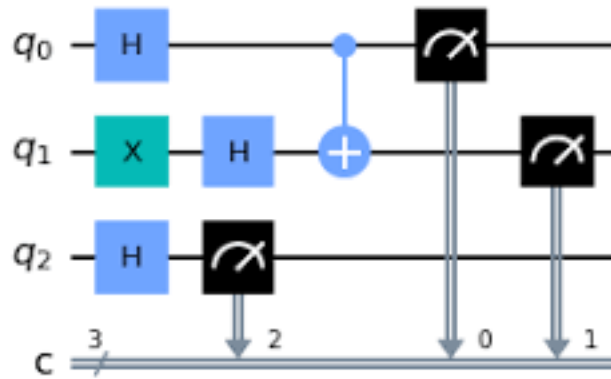
Hadamard gate :

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (1.3)$$

Rotation gates :

$$R_x(2\theta) = \begin{bmatrix} \cos(\theta) & -i \sin(\theta) \\ -i \sin(\theta) & \cos(\theta) \end{bmatrix} \quad R_y(2\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad R_z(2\theta) = \begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix} \quad (1.4)$$

A quantum algorithm that we use are some sets of quantum operations on the qubits followed by measurement, represented by a quantum circuit(Fig. 2)Figure 1.2, and then some post-processing of the measurement results.



**Figure 1.2:** An example of a Quantum Circuit

# Chapter 2

## Introduction to VQE

One of the leading quantum chemistry algorithms for use with near-term quantum computers is the Variational Quantum Eigensolver (VQE). A quantum computer is trained to prepare the ground state of a specific molecule in this application of the Quantum variational principle.[8]

The variational quantum eigensolver (VQE) is a technique for determining a Hamiltonian's eigenvalues using a hybrid quantum-classical computing methodology. Quantum phase estimation is one of the completely quantum algorithms that has been suggested as an alternative to VQE since QPE requires quantum hardware that won't be available anytime soon. The electronic Schrodinger equation has been successfully solved using VQE for a number of tiny molecules. The intricacy of the quantum circuits and the complexity of the conventional optimization issue, however, are what prevent this technology from scaling up. The variational ansatz used to describe the trial wave function has an impact on both of these elements. Consequently, the development of an effective ansatz is a current research topic. In other words, deep quantum circuits created by employing currently available ansatzes for problems that map into more than a few qubits cannot be executed by existing quantum computers.

A molecular Hamiltonian and a parametrized circuit that prepares the molecule's quantum state are the inputs to the VQE algorithm. The expectation value of the Hamiltonian calculated in the trial state is the definition of the cost function in VQE. By iteratively minimizing the cost function, the target Hamiltonian's ground state is

discovered. A classical optimizer performs the optimization using a quantum computer to analyze the cost function and determine its gradient at each optimization, a classical optimizer performs the optimization step.

Finding a series of quantum operations that prepares the lowest energy state (or minima) of an approximate representation of a target quantity or observable is the goal of the VQE. Even if the efficiency of estimating an observable's expectation values is the only strict condition for its representation, it is frequently the easiest if that operator has a compact or simple expression in terms of Pauli operators or tensor products of Pauli operators. It is frequently most practical to qubitize a fermionic system, which entails writing the system's many-body Hamiltonian using second quantization before employing a mapping to express the creation-annihilation operators in terms of Pauli operators. Jordan-Wigner transformation [7] is one of the often used fermion schemes.

Although these concepts appear appealing in principle, the processing power of the tiny quantum devices now in use, also known as NISQ (noisy intermediate-scale quantum) devices, is severely constrained. The inefficiency of current quantum algorithms, which are frequently addressed in terms of the quantum advantage, with regard to the resources required to solve any worthwhile issue on a quantum computer quicker than on a conventional computer is a second crucial factor. For instance, it is anticipated that the resources needed to do computations on the chromium dimer, a big enough chemical system to show the quantum advantage on a quantum computer, would require at least 1 million physical qubits, or 60 high-fidelity qubits. However, in order for error correction algorithms to function, these physical qubits must be more accurate than ones now in use. This is a long way from the physical quantum computers that are now accessible, and it will continue to be that way for a while.

## 2.1 VQE Overview

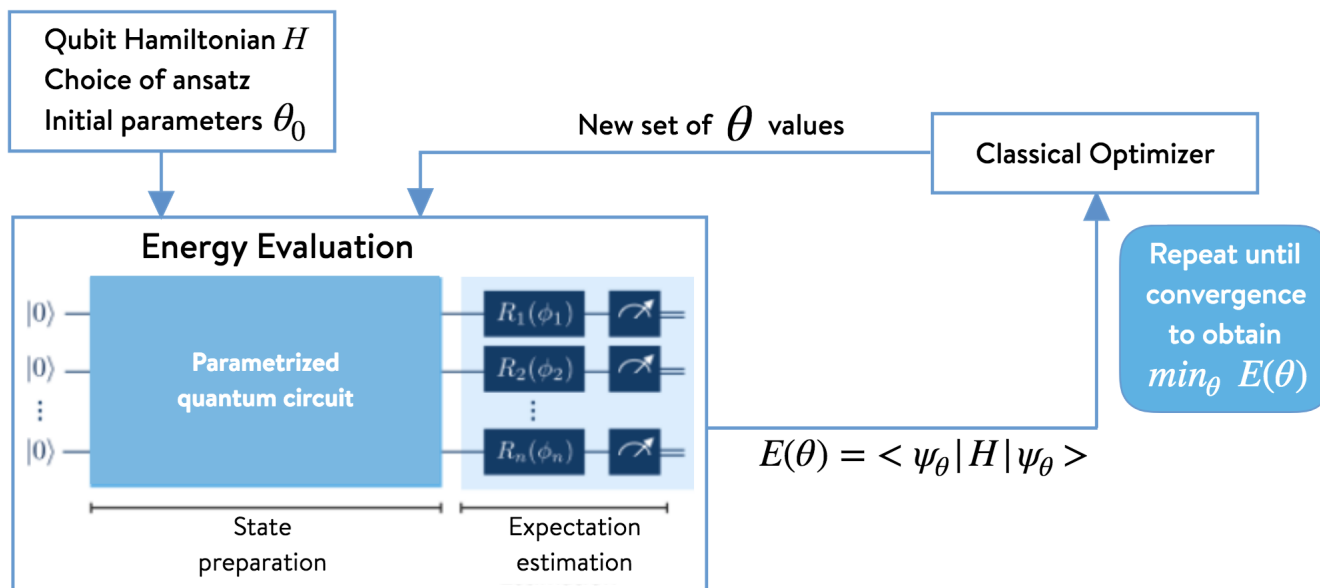


Figure 2.1: VQE Procedure flowchart

# Chapter 3

## Future Scope of VQE

The variational quantum eigensolver (or VQE), which Peruzzo et al. initially proposed in 2014, has drawn a lot of interest from the scientific community lately. It computes the ground state energy of a Hamiltonian, an issue that is crucial to quantum chemistry and condensed matter physics, using the variational principle. Due to the computational constraints placed on accurate modelling of the electronic wavefunction for these many-electron systems, conventional computing approaches are limited in their accuracy. One of the most intriguing near-term applications for quantum computing is the use of the VQE to describe these complicated wavefunctions in polynomial time. The fact that variational algorithms have some degree of tolerance to the noise in the quantum hardware is a significant benefit.

Finding a way to traverse the pertinent literature has quickly turned into an intimidating undertaking because there are so many approaches promising to improve various aspects of the algorithm but without detailed explanations of how the various components work together. The literature also extensively discusses the algorithm's possible practical benefits, although it comes to different conclusions. Strong theoretical foundations for great scaling of individual VQE components have been suggested, however tests have shown that their different pre-factors may be too high to achieve a quantum computing advantage over traditional approaches.

This project tries to deconstruct the pertinent literature to offer a thorough overview of the advancements achieved on the various components of the algorithm and to propose upcoming research topics that are essential for the VQE to live up to its potential. A thorough evaluation of the algorithm's various parts is conducted. These

include the representation of Hamiltonians and wavefunctions on a quantum computer, the search for ground state energies through optimization, the correction of quantum faults in the post-processing stage, and recommended best practises.

The paths for the VQE to attain quantum advantage as the quantum computing hardware grows up and as the noise levels are decreased will be determined by the responses to these open research issues.

Some researchers have attempted to estimate the tipping point for quantum computing-based quantum chemistry to overtake conventional methods. As one example, Elfving et al [4]. estimate the size of basis set (and hence the number of qubits) that would be required for a tangible quantum advantage of quantum computing based methods to lie somewhere between 19 and 34 molecular orbitals (or twice as many spin orbitals and hence twice as many qubits).

# Chapter 4

## Variational principle in Quantum Mechanics

The **variational technique** is one strategy to approximate the ground state with the lowest energy and some stimulated states in quantum mechanics. This makes it possible to calculate approximations of wavefunctions, like molecular orbitals. The variational concept serves as the method's foundation.

The technique involves selecting a "trial wavefunction" based on one or more parameters and identifying the values of these parameters that result in the energy expectation value being as low as possible. By setting the parameters to these values, a wavefunction that approximates the ground state wavefunction is produced, and the energy in that state's expectation value serves as an upper constraint on the ground state energy.

Assume we are given a Hilbert space and the Hamiltonian  $H$  (Hermitian operator). We analyse the discrete spectrum of  $H$  and a basis of eigenvectors called  $|\psi_\lambda\rangle$ , disregarding difficulties with continuous spectra.

$$\langle\psi_1|\psi_2\rangle = \delta_{12}, \text{ Where } \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \quad (4.1)$$

and  $[\psi_\lambda]$  satisfies the equation,

$$H|\psi_\lambda\rangle = \lambda|\psi_\lambda\rangle \quad (4.2)$$



Let's assume that the spectrum of  $H$  is constrained from below and that its highest lower bound is  $E_0$ , once more ignoring the complexities associated with a continuous spectrum of  $H$ . The value of  $H$ 's expectation in state  $|\psi_\lambda\rangle$  is then,

$$\begin{aligned} \langle \psi | H | \psi \rangle &= \sum_{\lambda_1, \lambda_2 \in \text{Spec}(H)} \langle \psi | \psi_{\lambda_1} \rangle \langle \psi_{\lambda_1} | H | \psi_{\lambda_2} \rangle \langle \psi_{\lambda_2} | \psi \rangle \\ &= \sum_{\lambda \in \text{Spec}(H)} \lambda |\langle \psi_\lambda | \psi \rangle|^2 \geq \sum_{\lambda \in \text{Spec}(H)} E_0 |\langle \psi_\lambda | \psi \rangle|^2 = E_0 \langle \psi | \psi \rangle. \end{aligned} \tag{4.3}$$

The lowest value would be  $E_0$ , and the corresponding state would be an eigenstate of  $E_0$ , if we were to vary across all feasible states with norm 1 while attempting to minimise the expectation value of  $H$ . Since physical computations typically require varying throughout the whole Hilbert space, a subspace of the complete Hilbert space is chosen and parametrized by some (real) differentiable parameters  $\alpha_i$  ( $i = 1, 2, \dots, N$ ). The ansatz is this selection of the subspace. The choice of ansatz is crucial since some choices produce better approximations than others.

Assume that the ansatz and the ground state overlap somewhat (otherwise, the ansatz is flawed). We want to make the ansatz normal following the constraint

$$\langle \psi_\alpha | \psi_\alpha \rangle = 1 \tag{4.4}$$

and we want to minimize

$$\epsilon(\alpha) = \langle \psi_\alpha | H | \psi_\alpha \rangle \tag{4.5}$$

In general,

We construct the functional for a hamiltonian  $H$  that characterises the investigated system and any normalizable function with arguments suitable for the unidentified wave function of the system :

$$\epsilon[\Psi] = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \quad (4.6)$$

According to the variational principle,

1.  $\epsilon \geq E_0$  where  $E_0$  is the ground state eigenvalue
2.  $\epsilon = E_0$  if and only if  $\psi$  precisely equals the ground state wave function of the examined system.

The variational approach used in quantum chemistry and quantum mechanics to locate approximations to the ground state is based on the variational principle[5] mentioned above.

# Chapter 5

## Nuclear physics to Quantum Computation problem

### 5.1 The Nuclear Hamiltonion

The ultimate objective of computational quantum chemistry is to fully understand the quantum effects that affect the structure and characteristics of molecules. Since the characteristic energies linked to these phenomena, such as the electronic correlation energy, are often a very small portion of the molecule's overall energy, achieving this aim is difficult.

The wave function that describes how the interacting electrons in a molecule interact may be used to calculate precise chemical characteristics. The Schrödinger equation is satisfied by the electronic wave function  $\psi(r)$ .

$$H_e\psi(r) = E\psi(r) \quad (5.1)$$

where  $H_e$  and  $E$  are the molecule's total energy and electronic Hamiltonian, respectively.

The Hamiltonian can be defined as :

$$\hat{H} = \hat{K}_N + \hat{K}_e + \hat{V}_{NN} + \hat{V}_{eN} + \hat{V}_{ee} \quad (5.2)$$

where we have decomposed the kinetic energy operator into nuclear and electronic

terms,  $K_N$  and  $K_e$ , as well as the potential energy operator into terms representing the interactions between nuclei,  $V_{NN}$ , between electrons,  $V_{ee}$ , and between electrons and nuclei,  $V_{eN}$ .

$$\begin{aligned}
\hat{K}_N &= - \sum_i^{\text{nuclei}} \frac{\hbar^2}{2M_i} \nabla_{\mathbf{R}_i}^2 \\
\hat{K}_e &= - \sum_i^{\text{electrons}} \frac{\hbar^2}{2m_e} \nabla_{\mathbf{r}_i}^2 \\
\hat{V}_{NN} &= \sum_i \sum_{j>i} \frac{Z_i Z_j e^2}{4\pi\epsilon_0 |\mathbf{R}_i - \mathbf{R}_j|} \\
\hat{V}_{eN} &= - \sum_i \sum_j \frac{Z_i e^2}{4\pi\epsilon_0 |\mathbf{R}_i - \mathbf{r}_j|} \\
\hat{V}_{ee} &= \sum_i \sum_{i<j} \frac{e^2}{4\pi\epsilon_0 |\mathbf{r}_i - \mathbf{r}_j|}
\end{aligned} \tag{5.3}$$

The nuclei of the molecule can be regarded as point particles with fixed coordinates when resolving the latter equation (Born-Oppenheimer approximation) [2]. Both the total energy and the electronic Hamiltonian are parametrically dependent on the nuclear coordinates in this approximation. So the Hamiltonian changes to a form;

$$\hat{H} = \hat{K}_e + \hat{V}_{NN} + \hat{V}_{eN} + \text{energy shift} \tag{5.4}$$

which then it becomes exactly,

$$\hat{H} = \sum_{i,j} \frac{1}{2} \langle i | \nabla_i^2 | j \rangle a_i^\dagger a_j + \sum_{i,j} \langle i | \frac{Z_A}{r_{iA}} | j \rangle a_i^\dagger a_j + \sum_{i,j,k,l} \langle i, j | \frac{1}{r_{ij}} | k, m \rangle a_i^\dagger a_j a_k a_m \tag{5.5}$$

This Hamiltonian is further simplified using the second quantization formalism which is explained below.

The second-quantization formalism, which we shall examine in further depth in the following section, is frequently used to express the electronic Hamiltonian of the molecule. A base of single-particle states must be selected to achieve this goal. These states, known as molecular orbitals in quantum chemistry, explain the wave function of a single electron in a molecule.

Ordinarily, atomic orbitals are combined in a linear fashion to represent molecular orbitals. Using the Hartree-Fock (HF) technique, the expansion coefficients in the atomic basis are determined. The HF approximation treats each electron in the molecule as a separate particle that travels in response to the nuclei's Coulomb potential and a mean field created by all the other electrons.

The second-quantized Hamiltonian can only be constructed using the optimised coefficients.

### 5.1.1 Second Quantization

The formalism known as second quantization, sometimes known as occupation number representation, is used to define and examine quantum many-body systems. Canonical quantization is a method used in quantum field theory in which the fields are viewed as field operators, much as how the physical variables (such as location, momentum, etc.) are viewed as operators in first quantization. The fields are often viewed as the wave functions of matter. Paul Dirac established the main concepts of this approach in 1927 [3]. According to this method, the quantum many-body states are represented in the Fock state basis, which is created by packing a specific number of identical particles into each single-particle state.

The creation and annihilation operators are introduced in the second quantization formalism to build and manage the Fock states, giving researchers studying the quantum many-body theory important tools.

### 5.1.2 Second Quantized fock states

Because the language of first quantization is redundant for indistinguishable particles, first quantized wave functions require challenging symmetrization processes to represent physically feasible many-body states. The many-body state is described

in the first quantization language by responding to a series of queries like "Which particle is in which state?" However, since the particles are similar, it is impossible to determine which particle is which in the first place, hence these are not physical problems. The quantum many-body states  $\psi_1 \otimes \psi_2$  and  $\psi_2 \otimes \psi_1$ , which appear to be distinct, are essentially the same thing under various labels. To get rid of this repetition in the initial quantization description, symmetrization (or anti-symmetrization) must be used.

$$|[n_\alpha]\rangle \equiv |n_1, n_2, \dots, n_\alpha, \dots\rangle \quad (5.6)$$

implying that there are  $n_\alpha$  particles in the single-particle state, which is represented by the display style  $|\alpha\rangle$  (or as  $|\psi_\alpha\rangle$ ).

$\sum_\alpha n_\alpha = N$  shows that the occupation numbers add up to the total number of particles. Because of the Pauli exclusion principle, the occupation number for fermions may only be 0 or 1, although for bosons it can be any non-negative integer.

$$n_\alpha = \begin{cases} 0, 1 & \text{fermions,} \\ 0, 1, 2, 3, \dots & \text{bosons.} \end{cases} \quad (5.7)$$

Fock states are another name for the occupation number states. The many-body Hilbert space, often known as the Fock space [1], is completely based on all of the Fock states. A linear collection of Fock states may be used to express any generic quantum many-body state.

## 5.2 Hamiltonian in action

In general the electronic hamitonion is written as :

$$H = \sum_{ij} h_{ij} a_i^\dagger a_j + \sum_{ijkl} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \quad (5.8)$$

Where  $h_{ij}$  and  $h_{ijkl}$  satisfy the needs of electron-nuclear repulsion term and electron-electron repulsion terms respectively.

Where,

$$h_{ij} \equiv \int dx \chi_i^*(x) \left( -\frac{1}{2} \nabla^2 - \sum_\alpha \frac{Z_\alpha}{r_{\alpha,x}} \right) \chi_j(x) \quad (5.9)$$

$$h_{ijkl} \equiv \int dx_1 dx_2 \left( \frac{\chi_i^*(x_1) \chi_j^*(x_2) \chi_k(x_2) \chi_l(x_1)}{r_{1,2}} \right) \quad (5.10)$$

( $h_{ij}$  is one-body integral and  $h_{ijkl}$  is a two-body intergral).

In this project we simulated Hamiltonian(for Deutron, there will not be  $h_{ijkl}$ ) term H written in second quantization form as [6] :

$$H = \sum_{ij} h_{ij} a_i^\dagger a_j \quad (5.11)$$

where  $h_{ij} = \langle i | \hat{T} + \hat{V} | j \rangle$  While  $a_i^\dagger a_j$  are assessed using quantum algorithms,  $h_{ij}$  are estimated using traditional methods. The Kinetic energy terms and the Potential terms make up the Hamiltonian for a straightforward nuclear system. The Kinetic Energy operator is given in Harmonic Oscillator Basis by:

$$\langle n'l' | \hat{T} | nl \rangle = \frac{\hbar\omega}{2} \left[ (2n + l + \frac{3}{2}) \delta_n^{n'} - \sqrt{n(n + l + \frac{1}{2})} \delta_n^{n'+1} - \sqrt{(n + 1)(n + l + \frac{3}{2})} \delta_n^{n'-1} \right] \delta_l^{l'} \quad (5.12)$$

where  $n(n) = 0, \dots, N - 1$  is the radial quantum number of the harmonic oscillator basis with  $N$  determining the size of the basis, and  $l(l')$  is the orbital angular momentum. For the case of the deuteron with  $l = 0$ ,

Equation 5.12 becomes:

$$\langle n'l' | \hat{T} | nl \rangle = \frac{\hbar\omega}{2} \left[ (2n + \frac{3}{2}) \delta_n^{n'} - \sqrt{n(n + \frac{1}{2})} \delta_n^{n'+1} - \sqrt{(n + 1)(n + \frac{3}{2})} \delta_n^{n'-1} \right] \quad (5.13)$$

The matrix element of potential terms for a Nuclear Hamiltonian depends upon the choice of potential, and for this project we consider,

**EFT Potential** : The matrix element for EFT potential is given by

$$\langle n' | v | n \rangle = V_0 \delta_n^{n'} \delta_n^0 \quad (5.14)$$

where the choice of coefficient  $V_0 = -5.68658111 \text{ MeV}$  for Deutron.

## 5.3 Mapping the Hamiltonian to physical Qubits

### 5.3.1 Jordan-Wigner Mapping

Note that the occupation basis, also known as the occupancy basis, is formed by using  $n$  qubits to store the occupation number of  $n$  electronic spin-orbitals. The associated  $i^{th}$  qubit is in the  $|1\rangle$  state if the  $i$ th molecular orbital is occupied. While the qubit is in the  $|0\rangle$  state if the molecular orbital is vacant.

We need to map fermionic operators onto operators that act on physical qubits which can be done using **Jordan-weigner** transformation [7].

Qubits are manipulated by quantum computers via Pauli matrix-based operations (denoted as  $X_q$ ,  $Y_q$ , and  $Z_q$  on qubit  $q$ ). The Jordan-Wigner transformation can transfer the deutron generation and annihilation operators onto Pauli matrices.

The transformations are as follows :

$$a_n^\dagger \rightarrow \frac{1}{2} \left[ \prod_{j=0}^{n-1} -Z_j \right] (X_n - iY_n) \quad (5.15)$$

$$a_n \rightarrow \frac{1}{2} \left[ \prod_{j=0}^{n-1} -Z_j \right] (X_n + iY_n) \quad (5.16)$$

Note that this mapping requires  $O(N)$  qubit operations to simulate one electronic operation.



# Chapter 6

## The Variational Quantum Eigensolver

### 6.1 Variational Ansatz

To use the variational technique on a quantum computer, the ansatz must be varied in a methodical way. Using a parameterized circuit with a fixed shape, VQE accomplishes this. Such a circuit is sometimes referred to as a variational form, and the linear transformation  $U(\theta)$  may be used to model its activity. A beginning state  $|\psi\rangle$  (such as the Hartree Fock state or the vacuum state  $|0\rangle$ ) is subjected to a variational form, which produces an output state  $|\psi(\theta)\rangle$ . An anticipated value of minimum is what iterative optimization over  $|\psi(\theta)\rangle$  seeks to produce. Although in reality effective bounds on expectation value

$$\langle\psi(\theta)|H|\psi(\theta)\rangle \approx E_{gs} \equiv \lambda_{min} \tag{6.1}$$

can be determined even if this is not the case,  $|\psi(\theta)\rangle$  will ideally be near to  $|\psi_{min}\rangle$  (where "closeness" is described by either state fidelity, or Manhattan distance).

Two competing objectives must be balanced while creating a variational form. Our  $n$  qubit variational form would ideally be able to create any feasible state  $|\psi\rangle$  where  $|\psi\rangle$  is an element of  $N = 2^n$  dimensional complex hilbert space. The variational form should, however, utilise as few parameters as feasible.

About the purpose of simplicity, we will ignore the second goal and instead focus on providing intuition for how to design variational forms that meet our first goal.

## 6.2 Parameter Optimization

Following the variational method's selection of an efficiently parameterized variational form, its parameters must be tuned to reduce the target Hamiltonian's expectation value. There are many difficulties in the parameter optimization procedure. For instance, the noise present in quantum hardware may affect the assessment of the objective function (energy calculation), which may not accurately reflect the goal function. Additionally, depending on the cardinality of the parameter set, certain optimizers carry out a number of objective function evaluations. The needs of an application should be taken into account while choosing a suitable optimizer.

Gradient descent is a well-known optimization technique where each parameter is updated in the direction resulting in the highest local change in energy. As a result, the quantity of evaluations carried out relies on the presence of optimization factors. As a result, the algorithm can scan the search space fast for a local optimum. The number of circuit evaluations required by this optimization approach is quite high, and it frequently stalls at subpar local optima. It is a simple optimization technique, however it is not suggested for VQE.

The classical optimizers used in this project is mainly, SPSA, SLSQP, and COBYLA. The Simultaneous Perturbation Stochastic Approximation optimizer (SPSA) is a suitable optimizer for optimising a noisy objective function. With just two measurements, SPSA may estimate the gradient of the goal function. In contrast to gradient descent, which perturbs each parameter separately, it accomplishes this by simultaneously perturbing all of the parameters in a random manner. The classical optimizer SPSA is advised when using VQE in a noisy simulator or on actual hardware.

Numerous classical optimizers may be helpful when noise is not present in the cost function evaluation (such as when utilising VQE with the statevector simulator). The Sequential Least Squares Programming optimizer (SLSQP) and the Constrained

Optimization by Linear Approximation optimizer are two examples of optimizers that Qiskit Aqua supports (COBYLA). It is important to note that COBYLA only does one evaluation of the objective function during each optimization iteration (and that the number of evaluations is thus independent of the cardinality of the parameter set). Therefore, it is advised to use COBYLA if the goal function is noise-free and decreasing the number of completed evaluations is desired.

### 6.3 Various Quantum Algorithms

Algorithm	Acronym
VQE	Variational Quantum Eigensolver
QFT	Quantum Fourier Transform
QPE	Quantum Phase Estimation
DJ	Deutsch Josza algorithm
QKD	Quantum Key Distribution
HHL	Harrow-Hassidim-Lloyd quantum algorithm
QOA	Quantum Optimization Algorithm
QAOA	Quantum Approximate Optimization Algorithm

# Chapter 7

## VQE Results for Deuteron

### 7.1 The Deuteron

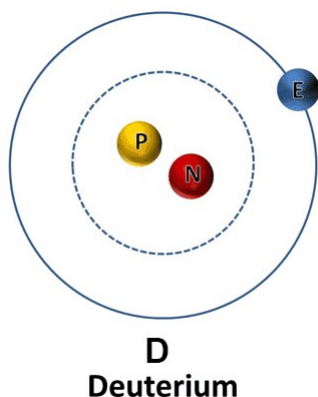


Figure 7.1: Deuterium Atom

Deuterium is one of two stable isotopes of hydrogen (the other being protium, or hydrogen-1). The nucleus of a deuterium atom, called a **Deuteron**, contains one proton and one neutron, whereas the far more common protium has no neutrons in the nucleus.

### 7.2 Simulator results

Reference value of ground state = -1.749

### 7.2.1 User defined Ansatz

1. **SLSQP** Results : -0.6921 for qasm simulator and -1.74916 from statevector simulator
2. **COBYLA** Results : -1.6341 for qasm simulator and -1.74916 from statevector simulator
3. **SPSA** Results : -2.012 for qasm simulator and -1.74916 from statevector simulator

### 7.2.2 QISKIT inbuilt Ansatz

1. **SLSQP** Results : +0.1908 for qasm simulator and -1.74916 from statevector simulator
2. **COBYLA** Results : -0.8438 for qasm simulator and -1.74915 from statevector simulator
3. **SPSA** Results : -1.7607 for qasm simulator and -0.1909 from statevector simulator

## 7.3 Why QASM and statevector simulators giving different results?

**statevector-simulator** keeps track of the state vector itself. When asking about the measurement probability, it would correspond to the absolute square of the amplitudes. **qasm-simulator** is similar to how things work on a hardware. You do not have access to the state vector, and you have to specify the number of shots. Each shot represents a sample in the measurement space. So the measurement probability would be the distribution of the samples.

The **QASM Simulator** is the main Qiskit Aer backend. This backend emulates execution of a quantum circuit on a real device and returns measurement counts. It includes highly configurable noise models and can even be loaded with automatically generated approximate noise models based on the calibration parameters of actual hardware devices.

The **Statevector Simulator** is an auxiliary backend for Qiskit Aer. It simulates the ideal execution of a quantum circuit and returns the final quantum state vector of the device at the end of simulation. This is useful for education, as well as the theoretical study and debugging of algorithms.

# Bibliography

- [1] Alexander Altland and Ben D Simons.
- [2] M. Born and R. Oppenheimer. Zur quantentheorie der molekeln. *Annalen der Physik*, 389(20):457–484, 1927.
- [3] Paul Adrien Maurice Dirac and Niels Henrik David Bohr. The quantum theory of the emission and absorption of radiation. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 114(767):243–265, 1927.
- [4] V. E. Elfving, B. W. Broer, M. Webber, J. Gavartin, M. D. Halls, K. P. Lorton, and A. Bochevarov. How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?, 2020.
- [5] L. D. Landau and L. M. Lifshitz. *Quantum Mechanics Non-Relativistic Theory, Third Edition: Volume 3*. Butterworth-Heinemann, 3 edition, January 1981.
- [6] Pooja Siwach and P. Arumugam. Quantum simulation of nuclear hamiltonian with a generalized transformation for gray code encoding. *Phys. Rev. C*, 104:034301, Sep 2021.
- [7] Andrew Tranter, Peter J. Love, Florian Mintert, and Peter V. Coveney. A comparison of the bravyi–kitaev and jordan–wigner transformations for the quantum simulation of quantum chemistry. *Journal of Chemical Theory and Computation*, 14(11):5617–5630, 2018. PMID: 30189144.
- [8] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, October 2019.

# Appendix A

## Ground state of Deuteron

Code for running in IBMQ simulators

```
import numpy as np
import pylab
from qiskit import *
from qiskit.utils import QuantumInstance, algorithm_globals
from qiskit.algorithms import VQE, NumPyMinimumEigensolver
from qiskit.algorithms.optimizers import SLSQP, SPSA, COBYLA, L_BFGS_B
from qiskit.circuit.library import TwoLocal
from qiskit.opflow import I, X, Z, Y
from qiskit.providers.aer import QasmSimulator

'''
This is the Hamiltonion of Deuterium molecule first created using
ladder operators then used dyson series and finally transformed to this form
using Jordan-wigner like methods. (Which is available online for free)
https://journals.aps.org/prc/pdf/10.1103/PhysRevC.104.034301
'''
H2_op = (5.906709 * I ^ I) + \
        (0.218291 * Z ^ I) - \
        (6.125 * I ^ Z) - \
        (2.143304 * X ^ X) - \
        (2.143304 * Y ^ Y)
```



```

#Classical eigen value calculation of the operator

npme = NumPyMinimumEigensolver()
result = npme.compute_minimum_eigenvalue(operator=H2_op)
# The classical solution to the problem

ref_value = result.eigenvalue.real
print(f'Reference value: {ref_value:.5f}')

# Creating user defined ansatz:

from qiskit.circuit import Parameter

theta = Parameter('theta')
ansatz_1 = QuantumCircuit(2)
ansatz_1.x(0)
ansatz_1.ry(theta, 1)
ansatz_1.cx(1,0)
ansatz_1.draw('mpl')

# 'TwoLocal' ansatz
'''
This is an ansatz creation method which is inbuilt in qiskit
'''
ansatz_2 = TwoLocal(rotation_blocks='ry',
                    entanglement_blocks='cz', reps = 1)

# Just for testing purpose
ansatz_2.draw('mpl')

#Using different combinations of optimizers and simulators
to compare for a given ansatz
optimizers = [COBYLA(maxiter=80),
              SLSQP(maxiter=60), SPSA(maxiter = 60)]
# maxiter is the maximum number of function evaluations

```

```

simulators = ['statevector_simulator', 'qasm_simulator']
# To verify the      results before going with Real backends

# Looping for the above lists and printing out
the results for 'ansatz_1' using VQE.

for i, optimizer in enumerate(optimizers):
    for j, simulator in enumerate(simulators):
        vqe = VQE(ansatz_1, optimizer,
                  quantum_instance=QuantumInstance(backend=Aer.get_backend(simulator)))
        result= vqe.compute_minimum_eigenvalue(operator=H2_op)

        print('For optimizer :', optimizer, 'and simulator : '
              , simulator, end      = '\n')
        print('Eigen value : ', result.eigenvalue, end ='\n')
        print('Eigen Value : ', result.eigenvalue, end ='\n')
        print('Optimal Value : ', result.optimal_value, end ='\n')
        print('Optimizer Time : ', result.optimizer_time, end ='\n')
        print(end = '\n\n\n')

```

### Results for Ansatz1 :

For optimizer : qiskit.algorithms.optimizers.cobyla.COBYLA and simulator : statevector simulator

Eigen value : (-1.7491611969643985+0j)

Eigen Value : (-1.7491611969643985+0j)

Optimal Value : -1.7491611969643985

Optimizer Time : 0.06676125526428223

For optimizer : qiskit.algorithms.optimizers.cobyla.COBYLA and simulator : qasm simulator

Eigen value : (-1.6341610019531272+0j)

Eigen Value : (-1.6341610019531272+0j)

Optimal Value : -1.6341610019531272

Optimizer Time : 4.25739860534668

For optimizer : qiskit.algorithms.optimizers.slsqp.SLSQP and simulator : statevector simulator

Eigen value : (-1.7491609341971766+0j)

Eigen Value : (-1.7491609341971766+0j)

Optimal Value : -1.7491609341971766

Optimizer Time : 0.0207979679107666

For optimizer : qiskit.algorithms.optimizers.slsqp.SLSQP and simulator : qasm simulator

Eigen value : (-0.6921057597656253+0j)

Eigen Value : (-0.6921057597656253+0j)

Optimal Value : -0.6921057597656253

Optimizer Time : 9.577551364898682

For optimizer : qiskit.algorithms.optimizers.spsa.SPSA and simulator : statevector simulator

Eigen value : (-1.7491612220155874+0j)

Eigen Value : (-1.7491612220155874+0j)

Optimal Value : -1.7491612220155874

Optimizer Time : 0.6750929355621338

For optimizer : qiskit.algorithms.optimizers.spsa.SPSA and simulator : qasm simulator

Eigen value : (-2.012054529296875+0j)

Eigen Value : (-2.012054529296875+0j)

Optimal Value : -2.012054529296875

Optimizer Time : 26.068110466003418

Ansatz 2 :

```
# Looping for the above lists and printing out the results
```

```
for 'ansatz_2' using VQE.
```

```
for i, optimizer in enumerate(optimizers):
```

```
    for j, simulator in enumerate(simulators):
```

```
        vqe = VQE(ansatz_2, optimizer, quantum_instance=
```

```
                QuantumInstance(backend=Aer.get_backend(simulator)))
```

```

result= vqe.compute_minimum_eigenvalue(operator=H2_op)
print('For optimizer : ', optimizer,
      'and simulator : ', simulator,end= '\n')
print('Eigen value : ', result.eigenvalue, end ='\n')
print('Eigen Value : ', result.eigenvalue, end ='\n')
print('Optimal Value : ', result.optimal_value, end ='\n')
print('Optimizer Time : ', result.optimizer_time, end ='\n')
print(end = '\n\n\n')

```

### Results for Ansatz2 :

For optimizer : qiskit.algorithms.optimizers.cobyla.COBYLA and simulator : statevector simulator

Eigen value : (-1.7491595305589742+0j)

Eigen Value : (-1.7491595305589742+0j)

Optimal Value : -1.7491595305589742

Optimizer Time : 0.6761023998260498

For optimizer : qiskit.algorithms.optimizers.cobyla.COBYLA and simulator : qasm simulator

Eigen value : (-0.8438329082031251+0j)

Eigen Value : (-0.8438329082031251+0j)

Optimal Value : -0.8438329082031251

Optimizer Time : 20.155860900878906

For optimizer : qiskit.algorithms.optimizers.slsqp.SLSQP and simulator : statevector simulator

Eigen value : (-1.7491611621259546+0j)

Eigen Value : (-1.7491611621259546+0j)

Optimal Value : -1.7491611621259546

Optimizer Time : 0.45404982566833496

For optimizer : qiskit.algorithms.optimizers.slsqp.SLSQP and simulator : qasm simulator

Eigen value : (0.19085928320312373+0j)

Eigen Value : (0.19085928320312373+0j)

Optimal Value : 0.19085928320312373

Optimizer Time : 165.09524774551392

For optimizer : qiskit.algorithms.optimizers.spsa.SPSA and simulator : statevector simulator

Eigen value :  $(-0.19090278673387137+0j)$

Eigen Value :  $(-0.19090278673387137+0j)$

Optimal Value :  $-0.19090278673387137$

Optimizer Time : 1.391674518585205

For optimizer : qiskit.algorithms.optimizers.spsa.SPSA and simulator : qasm simulator

Eigen value :  $(-1.7607981054687514+0j)$

Eigen Value :  $(-1.7607981054687514+0j)$

Optimal Value :  $-1.7607981054687514$

Optimizer Time : 75.3864676952362

# Appendix B

## QISKIT version details

'qiskit-terra': '0.22.2', 'qiskit-aer': '0.11.1', 'qiskit-ignis': None,  
'qiskit-ibmq-provider': '0.19.2', 'qiskit': '0.39.2', 'qiskit-nature': '0.5.0',  
'qiskit-finance': '0.3.4', 'qiskit-optimization': '0.4.0', 'qiskit-machine-learning':  
'0.5.0'