Application of Quantum Computing in Physics

M.Sc. Thesis

By

Garvit Shrivastava

Department of Physics
Indian Institute of Technology Indore
June 2022

Application of Quantum Computing in Physics

A Thesis

Submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Science

By

Garvit Shrivastava

Department of Physics
Indian Institute of Technology Indore
June 2022

Indian Institute of Technology Indore

Candidate's Declaration

I hereby certify that the work which is being presented in thesis entitled **Application of Quantum Computing in Physics** in the partial fulfillment of the requirements for the award of the degree of **Master of Science** and submitted in the **Department of Physics, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July 2021 to June 2022 under the supervision of **Dr. Ankhi Roy, Associate professor, Indian Institute of Technology Indore.**

The matter presented in thesis has not been submitted by me for the award of any other degree of this or any institute.

Garvit
05/06/2022
Signature of student with date
(Garvit Shrivastava)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Ankhi Ray **05/06/2022**
Signature of the Supervisor of
M.Sc. thesis (with date)
(**Dr. Ankhi Roy**)

Garvit Shrivastava has successfully given his M.Sc. Oral Examination held on **03 June 2022**.

Ankhi Ray
Signature of supervisor of MSc thesis
Date: **05/06/2022**

Convener, DPGC
Date: 06/06/2022

Signature of PSPC member no. 1
Date: 06/06/2022

Signature of PSPC member no. 2
Date: 06/06/2022

Acknowledgement

First and foremost, I would like to express my deep gratitude towards my supervisors, Dr. Ankhi Roy, Associate professor, Indian Institute of Technology Indore without her guidance and wavering support it would have been impossible to get a better understanding of the subject. It has been a huge privilege to work under her for my thesis and it goes without saying how much her knowledge, supervision has helped me grow. She mentors, gave the most invaluable suggestions and boosted my morale whenever I faced any kind of dispiriting hurdles and for that I am indebted to her forever.

To have Dr. Subhendu Rakshit and Dr. Manavendra Mahato as my PSPC members has been the most humbling experience ever. Their inspiration and motivation has played a major role in the development of this thesis and for that I am extremely grateful to both of them.

I would like to show my thankfulness to Dr. Sudeep Ghosh, IBM Quantum, Bengaluru, India. He was always available for any discussion or guidance that I need.

Dedicated to my parents.

Abstract

Quantum computing is based on quantum mechanics and its phenomena. It promises to provide high computational power, high speed compare to classical computers and solve unsolvable problems of classical computers. Starting with the basics of quantum computing and its advantages, application of quantum computing in physics is the focus of this thesis. As application of computers, Machine learning systems are increasingly reaching the boundaries of classical computational models as the amount of data continues to grow. In this way, quantum computing power can help with machine learning problems. Quantum machine learning is the study of how to design and deploy quantum software to enable machine learning that is faster than that of classical computers. Many body problem can be better understood by simulating on computers. Classical computers are not that efficient for this task. Quantum simulators would allow researchers to investigate new physical phenomena by tackling this issues. This thesis describes our work of using quantum computing for experimental high energy physics data analysis and simulation of atomic nuclei.

Contents

List of Figures

Chapter 1

Introduction

The classical computer works on the laws of classical physics and mathematical logic and the advancement of classical computer is connected with miniaturization of integrated circuit. For more processing power, the number density of transistors in integrated circuit increases. As stated in Moore's law, the number density of transistors on an integrated circuit doubles in every 18 months or two years. Increasing number density will take the size too small and spacing between two component will go to atomic dimension where classical physics will not operate. Also because of this spacing between components keep on decreasing, there will be a heating problem and it may affect and damage the nearby components. Then Richard Feynman published an article entitled *"Simulating Physics with Computers"* [1], where he asked if quantum physics can be simulated on computer, and proposed a idea of using quantum computers to simulate quantum system. The principles of quantum physics inspired the concept of quantum computers.

Obtaining a factor of a large number on classical computers is computationally expensive. In 1994, Peter Shor developed a quantum mechanics-based factoring algorithm for quickly determining the prime factor of a large number. To speed up the process, the principle of quantum mechanics is used and it has started the era of quantum computers.

Information in classical computers is stored in bits. A classical bit can be in any two of the states, either in 0 or in 1, at a given time. However the quantum bit can be in 0 or 1 or both simultaneously. For 2 bits registers in classical computers, there are four combinations of states 00,01,10 and 11 in which a two bit register can be, whereas in quantum computers, a two bits register can be in any of these

four states and simultaneously in linear combination of these four states. So, the advantage of having more bits will speed up the process.

Classical computers work in a serial fashion. It completes a task and then moves on to the next one. If there is a problem with set of independent logics that need to be executed simultaneously. The mechanism called parallel computing is used to solve such types of problems, where n-bits are used to do n number of jobs simultaneously. However quantum computers allows us to use multiple bit in inherent parallelism. The reason to use quantum computing rather than parallel computing is that while using n-qubits in quantum computers, the linear combination gives us a quantum advantage to speeds up the process. This linear combination can use all the possible states in quantum registers.

Classical computers use logic gates as operation. These operations are irreversible processes and according to Landauer principle, operations transform energy into heat. But in quantum computers, the gate we use as operations are reversible, so the energy loss in the quantum process is negligible as compared to classical. The major problem in quantum computers is that in qubit measurement the result will be probabilistic in nature because a state in quantum mechanics is a linear combination of certain states. So during the measurement, it can measure any of the state out of them. To overcome this problem, measurements were done multiple times and statistics has been used.

Building a hardware of quantum computers is also a challenging task. Due to the heat and light, the stored information can be destroyed as well as quantum bit can lost their quantum properties. Longer coherence time and number of quantum bit are also challenging. Quantum errors like quantum bit flipping and extra added phase are also a challenging part. The error part has to be reduced and head towards the fault tolerant quantum computer. Till now 127-qubit quantum computer [2] has been made and there is a road-map to achieve the 1000-qubit in next 10 years with less error or noise. IBM had developed first quantum computer using superconductivity. IBM has provided a platform named IBM Quantum Experience where one can work with quantum simulator and quantum real hardware. In this thesis work, IBM Quantum Experience has been used to execute our code.

In high energy physics there is a large amount of data for particle identification. Classical computers are used to analyse this data using machine learning techniques. However classical computers do it efficiently, but Quantum computers can be used for this analysis. Quantum computers and machine learning can tackle the large

data in high energy physics and intersection of machine learning and quantum computing is known as quantum machine learning. In this thesis we try to implement quantum machine learning on high energy physics data to improve the results.

Many body problem in physics has many interactions and complex structures. Simulation of many body on classical computers is difficult task but Quantum computers may provide a way to simulate it. Simulating a system using quantum computers will provide more computational power than classical simulation. Quantum simulation is a type of simulation that uses quantum computing. In this thesis we try to simulate deuteron as two body system and it will lead to simulate few body and then many body system on quantum computers.

This thesis is organised as follows: first we introduce the basics of quantum computing and the methodology to create a quantum circuit and it's measurement. In next chapter, we discuss about quantum machine learning and its implementation on experimental high energy physics data. Then in next chapter, we introduce a quantum simulation and discuss how it can be useful for many body physics problems. Finally, we discuss the results of quantum computations preformed on IBM Quantum.

Chapter 2

Basics of Quantum Computing

Every computation requires three basic elements: data, operation, and results. Data is stored in binary digits (0 and 1) in classical computers. In classical computers, the operation is followed by the well known logic gates, and the results are stored in binary digits. These binary numbers can be translated to a language that can be displayed on a computer screen. In the case of quantum computer, the input data for quantum computing is in the form of qubits. Quantum gates are used for operations, and results are in the form of measurements. In this chapter, we'll talk about these aspects of quantum computing and quantum mechanical properties such as : superposition, entanglement and interference, which are used for quantum advantages.

2.1 Qubit

As we have discussed, that classical computers work on classical bits. These bits store information based on voltage (high or low voltage) or charge (plus and minus) or spin (up and down), but in quantum computers we use quantum bits (qubit). These quantum bits stores information based on direction of electron spin. In classical, the bits can be in any state either 0 or 1, whereas in quantum the qubit can be in state of $|0\rangle$ or $|1\rangle$ or a linear combination these state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with the condition that α and β are such that $|\alpha|^2 + |\beta|^2 = 1$. This representation of qubit is in Dirac notation form and we can write a vector in a matrix form. So

the state $|0\rangle$ and $|1\rangle$ can be written as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$|\psi\rangle$ can be written as

$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

The linear combination of state $|0\rangle$ and $|1\rangle$ is called the superposition state, where a qubit can be in both of the state simultaneously.

There are different ways to represent a qubit, one of them is Bloch sphere. A Bloch sphere is a unit radius sphere and a qubit lies on the surface of this sphere. North pole of this sphere is represented as $|0\rangle$ and south pole is represented as $|1\rangle$. A superposition state can be represented in between of these poles. In Bloch sphere, poles are in the z direction and diameter of sphere is in xy-plane. A state $|\psi\rangle$ can be represented on Bloch sphere as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{\iota\phi}\sin\frac{\theta}{2}|1\rangle$$

Here angle θ is along with the z-axis (polar angle) and ϕ is angle from x-axis formed when state is projected onto xy-plane (azimuthal angle).

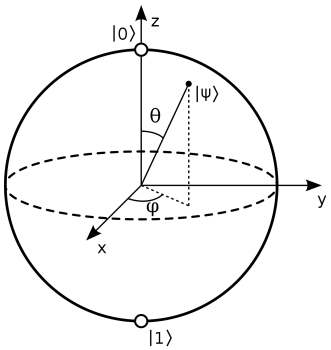


Figure 2.1: Bloch Sphere representation

Two qubits register in quantum computers can be in $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ states and in linear combination of these four states. We can write first four possible

states in the matrix form as

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and the linear combination of these four state can be written as in generalised way to represent two qubits as

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

The condition for these four complex coefficients is given as $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. All of these states come from the basis states $|0\rangle$ and $|1\rangle$, and higher number of qubits can be represented in terms of cross product (kronecker product) of these basis states $|0\rangle$ and $|1\rangle$ and their superposition state. Two qubits system has entanglement property that will discussed in next section as it requires the knowledge of quantum gates as well. A more generalized way to represent a n-qubit system is

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + ... + \alpha_{2^n-1}|2^n - 1\rangle.$$

Here $\sum_i |\alpha_i|^2 = 1$ and state we have written in decimals instead of binary. When we use this form of qubit, we need to use binary numbers inside bra-ket notation. For n bits, classical computers can have only one value out of 2^n possible permutations, while for n qubits, quantum computers can have all 2^n possible permutations.

2.2 Quantum Gates

The processing of these information in classical computers is carried out by logic gates. As classical computers use binary codes (bits), so the operations are defined by Boolean algebra. We are familiar with these logic gate operations. Whereas in quantum computers information processing is carried out by quantum logic gates. In quantum computers, quantum bits are used so operations are defined by linear algebra and can be represented by unitary matrices with complex elements [3]. For an operation in quantum computing, we use unitary transformation. Classical computers have AND, OR, NOT gate, whereas in quantum computers we have X, Z, H and other rotation gates which are unitary matrices. Quantum gates are qubit dependent like single qubit gate and two qubits gate. We can create multi-qubit quantum gates using single qubit gates. First let us talk about single qubit gate, it is 2×2 matrix. We have pauli operators $(\sigma_x, \sigma_y, \sigma_z)$ as gates in quantum computing and their matrix representation is given as,

$$\mathrm{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathrm{Y} = \begin{bmatrix} 0 & -\iota \\ \iota & 0 \end{bmatrix} \quad \mathrm{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

It can also be written as Dirac notation form

$$\mathrm{X} = |0\rangle\langle 1| + |1\rangle\langle 0|$$

$$\mathrm{Y} = -\iota|0\rangle\langle 1| + \iota|1\rangle\langle 0|$$

$$\mathrm{Z} = |0\rangle\langle 0| - |1\rangle\langle 1|$$

These gates are nothing but a representation of rotation around the Bloch sphere, X gate is rotation on Bloch sphere around x-axis by angle π and similarly Y and Z gates are also rotation around y- and z-axes respectively by an angle π. If we talk about there operations on qubit, we can see from matrix multiplication that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. These quantum gates basically flip our qubits and it is obvious as the qubits (0 and 1) are separated by an angle π and rotation of these gates are also by an angle π. This X gate is like NOT gate for quantum computing when working on z basis. Y and Z gate add extra phase on these basis states of qubit. This phase is global phase and we can not measure it. Now we talk about the most important, Hadamard gate. It is represented by H and creates a

superposition of qubit in single qubit system. H gate is rotation around y-axis by angle $\pi/2$ followed by rotation around x-axis by an angle π. Matrix representation and Dirac notation form of H gate is

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

$$\mathrm{H} = \frac{1}{\sqrt{2}}|0\rangle\langle 0| + \frac{1}{\sqrt{2}}|0\rangle\langle 1| + \frac{1}{\sqrt{2}}|1\rangle\langle 0| - \frac{1}{\sqrt{2}}|1\rangle\langle 1|$$

If H gate is applied on basis state then

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

$$\mathrm{H}|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

$|+\rangle$ and $|-\rangle$ are x basis states and these are eigenstates of pauli matrix. If H gate is applied on z basis state then we get $\mathrm{H}|+\rangle = |0\rangle$ and $\mathrm{H}|-\rangle = |1\rangle$, so this H gate changes the basis like from x basis to z basis and vice-versa. $|+\rangle$ is superposition state of $|0\rangle$ and $|1\rangle$, and if H is applied then it will collapse into $|0\rangle$ state as constructive interference whereas probability of measuring $|1\rangle$ is zero now, it is known as destructive interference. Interference allow us to bias measurement of qubit towards desired state. There are rotational gates also in quantum computing which describe the rotation around the axis by an angle θ, named as R_x, R_y and R_z. For fix rotation around z-axis we have S and T gate for rotation by $\pi/2$ and $\pi/4$ respectively. There is a generalized way to write a quantum gate in terms of unitary matrix with three parameters.

$$\mathrm{U}(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\theta/2) & -e^{\iota\lambda}\sin(\theta/2) \\ e^{\iota\phi}\sin(\theta/2) & e^{\iota(\phi+\lambda)}\cos(\theta/2) \end{bmatrix}$$

Here θ, ϕ and λ are angles. By adjustment of these parameters we can create any quantum gate, for example $\mathrm{U}(\pi/2, 0, \pi) = \mathrm{H}$. For 2 qubit, the famous Controlled-NOT (CNOT) gate which uses two qubits (one is control qubit and another is target

qubit) has matrix form as

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

As if we control 0 state then it will not affect target qubit but if we control 1 state then it will apply X gate on target qubit and control qubit does not change for it. This can be shown as,

$$\text{CNOT}|00\rangle = |00\rangle \text{ and } \text{CNOT}|01\rangle = |01\rangle$$

$$\text{CNOT}|10\rangle = |11\rangle \text{ and } \text{CNOT}|11\rangle = |10\rangle$$

Other control rotation gates are also there in two qubit system. For n-qubit, quantum gate is $2^n \times 2^n$ matrix. Now we can talk about entanglement. Entanglement of qubit happens when two qubits are entangled in such a way that one can not write it as combination of two separate qubit state and measurement of one qubit will collapse the other one in same state. Entangled qubit has to be measured together, it destroys the information while measured independently. Entangled state can be created by applying hadamard gate on first qubit followed by CNOT gate on second qubit (control on first qubit). A sample quantum entanglement circuit of two qubits is shown in figure 2.2. We have initialized both quantum bits register at $|0\rangle$ state.

$$|00\rangle \xrightarrow{\text{H}} \frac{(|0\rangle + |1\rangle)|0\rangle}{\sqrt{2}} = \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

This is one of the bell state out of four and state after CNOT in this quantum circuit (figure 2.2) is bell state that has derived above.

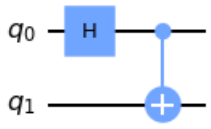


Figure 2.2: Quantum Entanglement circuit

2.3 Measurement

The concept of measurement is one of the factors that distinguishes quantum computing from classical computing. Measurement is an operator in quantum mechanics and when we take such measurements of system, then the system will collapse to one of its eigenstate. After measurement the quantum state will drop down to the definite single value. Measurement is an irreversible process because it to assigns measured quantum state into a single value. Since, measurement is probabilistic in nature so, if the numerical value is a then it should be $1 \geq |a|^2 \geq 0$. Measurement of 0 and 1 in matrix form can be written as

$$M_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad M_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

. When we take a measurement of state $|\psi\rangle$ in $|0\rangle$ state then it can be written as $\langle\psi|M_0|\psi\rangle$ and measurement of state $|\psi\rangle$ in state $|1\rangle$ written as $\langle\psi|M_1|\psi\rangle$. measurement of single qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in $|0\rangle$ state is

$$\langle\psi|M_0|\psi\rangle = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

This will give numerical value $|\alpha|^2$ or probability of measuring $|\psi\rangle$ in state $|0\rangle$. Similarly for measurement of $|\psi\rangle$ in state $|1\rangle$ have probability of $|\beta|^2$. From completeness property $\langle\psi|\psi\rangle = 1$, so $|\alpha|^2 + |\beta|^2 = 1$ indicates that the sum of all probability is 1. Measurement is basis dependent and we measure in z basis as we are working on z basis. Measurement of one register of entangled state is not possible because a entangled state can not be described as a tensor product of two qubit. If we take the measurement of one register of entangled state then it will affect the state of other register. When the measurement of all register that was entangled is taken, then by the affect of entanglement the other register collapse one register's state. Bell state will measured as if we measure one of the qubit to be $|1\rangle$, then the other qubit must also be $|1\rangle$ because combined state is $|11\rangle$.

Chapter 3

Quantum Machine Learning

Machine learning (ML) involves computer to learn without being explicitly programmed. ML is an ability of computer to learn. We have to feed sample data, the ML algorithm will learn the pattern from this data and predict the output. In experimental high energy physics, we have a large amount of data, here ML comes into picture. ML can be used for particle identification and regression. Classical machine learning is giving satisfactorily results in high energy physics data and is efficient to handle the data as of now. We are trying to use Quantum Machine Learning (QML) in high energy physics to check if it is better in comparison to classical ML or not. Most of the time we use Quantum Machine Learning as analysis of classical data on quantum computers. We use hybrid method that involve both classical and quantum computers. For quantum information processing, we have to encode a given classical dataset into a quantum state. This can be done by Feature Map. A feature map can transform data into a space where it is easier to process [4]. In general it transforms data into Hilbert space. The workflow of QML is shown in figure 3.1. In classical machine learning we have input features whereas in quantum machine learning features are encoded into quantum state and each feature is equal to a qubit. Then we can use quantum algorithms on encoded data and after the operations we have a final state. Results of quantum computation are read out by measuring.

To use these quantum machine learning algorithms, we are using a Compressed Baryonic Matter (CBM) experiment dataset which is about the muon identification from hadrons. We have to find rare signal events from the background with machine learning process. If the particle is muon then it is a signal event or if it is other

particle than muon then it will be a background event.

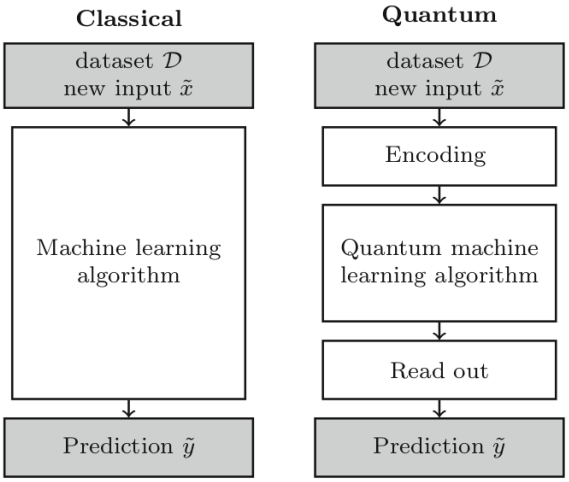


Figure 3.1: Quantum Machine Learning Workflow [4]

3.1 Classical Machine Learning

Before going into quantum machine learning let us first briefly introduce with classical machine learning. The Machine learning algorithms create a mathematical model with the use of sample historical data, known as training data, that helps in making predictions without being explicitly programmed. Machine learning combines computer science and statistics to create models. Algorithms that learn from past data are used in machine learning. We feed training data into a machine learning model as input, and the algorithm learns the pattern and characteristics from this data. Following this training, we will feed our model with new data, which we will refer to as testing data. Our model will predict the label of this new data based on previous data learning. The more information or training data we supply, the better our model performance will be. The model we've picked has an impact on this prediction. There are two types of machine learning models: supervised and unsupervised. In supervised learning, sample labeled data is provided to the model, and the algorithm operates with the supervision of this data, whereas in unsupervised learning, the data is not labeled or categorized, and the algorithm operates without supervision. There are two types of algorithms in supervised learning: classification and regression. Classification predicts the desecrate class labels as true and false whereas regression predicts continuous quantity or value such as price and

age. In our experimental dataset, We need to identify whether the particle is a muon or not, so this dataset is binary classification data, and we apply a binary classification machine learning algorithm to this data. Support Vector Machine (SVM) has provided us the excellent results in classification tasks in HEP in recent years [5]. For classification, we will use SVM in our work

The goal of SVM is to find the optimal separation hyperplane to separate two distinct classes. SVM selects the hyperplane with the help of extreme points and these points or vectors, support the hyperplane. That is why we call them as support vectors and algorithm is named after them as the Support Vector Machine. The margin of separation between two classes is determined by support vectors and maximized by the algorithm. The dimension of hyperplane depends on the number of features, if there are 3 features then there will be a 2-dimensional plane. A separating hyperplane in linear separation problem can be described in mathematical form as $\boldsymbol{w}^T.\boldsymbol{x} + b = 0$. Training data points are either above or below this plane and w and b can be chosen for a support vectors as $\boldsymbol{w}^T.\boldsymbol{x}_k + b = \pm 1$. An equality will be satisfied for correctly classified testing points or vectors when we multiply it by y as : $y_i(\boldsymbol{w}^T.\boldsymbol{x}_i + b) - 1 \geq 0$ and the separation margin given by ρ as : $\rho = \frac{2}{|\boldsymbol{w}|}$ as shown in figure 3.2.

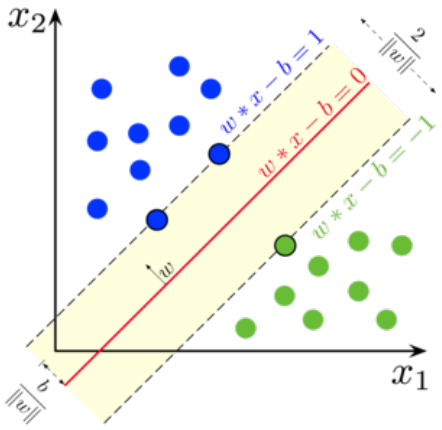


Figure 3.2: Support vector machine

By minimizing $|\boldsymbol{w}^2|$, we can maximize the margin of hyperplane from support vectors. This problem can be solved using Lagrange multipliers α_i as

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

and from this relation, we can estimate the boundary condition

$$\boldsymbol{w}^T.\boldsymbol{x} + b = \sum_i \alpha_i y_i (\boldsymbol{x}_i^T.\boldsymbol{x}) + b$$

and the prediction can be make as +1 and -1 or into two classes using signum function for new data $\boldsymbol{x}$ as

$$y = sgn\left[\sum_i \alpha_i y_i (\boldsymbol{x}_i^T.\boldsymbol{x}) + b\right]$$

The dual representation of our problem is

$$\mathcal{L} = \frac{1}{2}\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T.\boldsymbol{x}_j$$

We have to find Lagrange multipliers to maximize this Lagrangian. There are conditions that $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$. After getting the Lagrange multiplier we can use the above mention formula to get w and decision boundary. This is the solution for linear separation problem. The problems in which the distributions are overlapping are solved by the kernel trick. We go for higher dimensions to find the separating hyperplane with maximum margin easily, this is known as kernel trick. This kernel trick is useful to reduce calculation for non-linear separation problems. We transom our data into higher dimensional feature space and maps data points from x to $\phi(x)$ here ϕ is the basis function. But it is computationally expensive to calculate in higher dimension. So we will use the kernel function k(x,z) which is inner product of $\phi^T(x)$ and $\phi(z)$ in transformed space, we can skip the complex calculation part and use Lagrange multipliers to maximize the margin for hyperplane. Now the dual representation in this problem is

$$\mathcal{L} = \frac{1}{2}\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi^T(\boldsymbol{x}_i).\phi(\boldsymbol{x}_j)$$

Here the dot product of mapped data is verified as kernel function $k = \phi^T(\mathbf{x}_i).\phi(\mathbf{x}_j)$ The decision boundary can be estimated from

$$\sum_i \alpha_i y_i \phi^T(\mathbf{x}_i).\phi(\mathbf{x}) + b$$

and the prediction for classification is given by

$$y = sgn\left[\sum_i \alpha_i y_i(\phi^T(\mathbf{x}_i).\phi(\mathbf{x})) + b\right]$$

In SVM, different types of kernels are presented depending on the kernel function. Polynomial, Sigmoid, Radial basis function (RBF), and other kernel functions are there in SVM. As from accuracy point of view we have observed that RBF kernel gives best results. Accuracy in ML can be calculated by dividing the number of correct prediction by the total number of samples. So, RBF kernel gives best results and it is not easy to beat but we try to achieve the best we can from quantum computing.

3.2 Feature Map

Before going into the quantum machine learning we should know that we have to use data as in quantum state form and mapping this data from classical to quantum states is called preprocessing of quantum machine learning. This process is more important because we can not feed classical data directly to our quantum model. This can be done by feature map that has ability to encode data into quantum state. There are different methods to encode the data points into a quantum state. Basis encoding and amplitude encoding are two forms of encoding that will be discussed here and other methods of encoding can be found in [4]. In basis encoding, we use basis and superposition of basis state, we firstly convert the data into binary digits and arrange it in binary sequence. Using Dirac notation the binary sequence can used as qubit. For example, we have to encode data point $x(0.1, -0.6, 1.0)$. At first we convert decimal number into binary digits

$$0.1 \equiv 0\ 0001$$

$$-0.6 \equiv 1\ 1001$$

$$1.0 \equiv 0\ 1111$$

Binary sequence of them is 00001 11001 01111 and a quantum state can be written as

$$|00001\ 11001\ 01111\rangle$$

As it requires more number of qubits, so we approach amplitude encoding which is most commonly used. Amplitude encoding encodes the information as amplitude of a quantum state. For example if we encode the same data point $x(0.1, -0.6, 1.0)$. Firstly the points value has to be normalized in 2^n values, here n=2. The normalized point $x' = (0.073, -0.438, 0.730, 0.000)$. These values are used as amplitude of quantum states and can be represented by 2 qubit as

$$0.073|00\rangle - 0.438|01\rangle + 0.730|10\rangle + 0.00|11\rangle$$

Feature map transform data point $x \rightarrow |\Phi(x)\rangle$ by unitary transformation $\mathcal{U}_\Phi(x)$. A feature map is an arrangement of quantum gates in such a way that it can create desired quantum states and it is variational circuit whose parameter is dependent on input data. A simple model of feature map is presented in figure 3.3 where H gate creates superposition of every qubit then $\mathrm{U}_\Phi(x)$ encodes classical variables to quantum states by applying a rotational gates (R_z, R_y) of an angle x_i and CNOT gate uses for entanglement of qubits. Depth of a feature map is a repetition of quantum circuits (operation gates). We also have predefined feature maps and we can create it too. There are different types of predefined feature maps available named as Z Feature Map, ZZ Feature Map, Pauli Feature Map, Raw Feature Vector. In our work we have used Pauli Feature map. We have chosen this feature map from hyper-parameter tuning which is a method of trying different approach with variables and selecting the best one out of them.Encoding the data points is not the only work that feature maps do. As in classical ML, we use the dot product of feature vectors. We use feature map to compute the kernel function in quantum circuit.

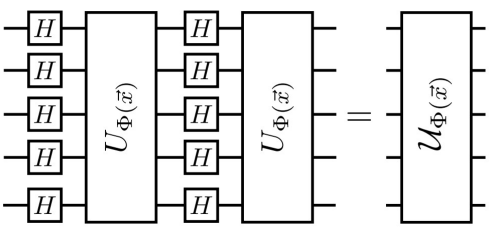


Figure 3.3: Feature Map [6]

3.3 Variational Quantum Classifier

Variational Quantum Classifier (VQC) is a classification algorithm in quantum computing. It has analogy from classical SVM and we use variational quantum circuit to find the hyperplane. As from its name it use variational parameterized circuit and use optimization for optimum results. Like classical machine learning, VQC also has a training stage where the training data points are provided and learning takes place. On testing stage where the data points without labels are provided which are then classified. VQC has a four stage process as in figure 3.4. First we use a feature map to load the data into a quantum system that allows effective embedding of classical data into the quantum system. After this in the second step we have a variational circuit parameterized by angle θ. The parameters of this variational circuit are then trained in the classical optimization loop until it classifies the data points correctly. This is the classical loop that trains our parameters until the cost function's value decreases. It is essential to choose a variational circuit of shorter depth to make it executable on real quantum hardware and number of parameterized quantum circuits (PQC) is depth here. In PQC we use rotation gates like parameterized R_y gates and CNOT gates. In the third step we take measurement of the quantum circuit. An n bit classification string is obtained. Once we measure the variational quantum circuit using the Measurement operator, the n bit string is now assigned values of the classification classes. This is done with the help of a boolean function $f : 0, 1^n \rightarrow 0, 1$, the analogy from svm is used here as with PQC, and the boolean function, which is the diagonal operator, is measured, giving us the value of $w.x$, which is then utilized to label [6]. In the fourth step we use the classical optimization loop. The parameters of the quantum variational circuit are updated using a classical optimization routine once the measurements are ready. We change the number of iterations to find global minima with optimizer. There are types of different optimizer there each with different method, here we use COBYLA optimizer from hyperparameter tuning. After all iterations we have the final output and we can get accuracy score and metric from here.

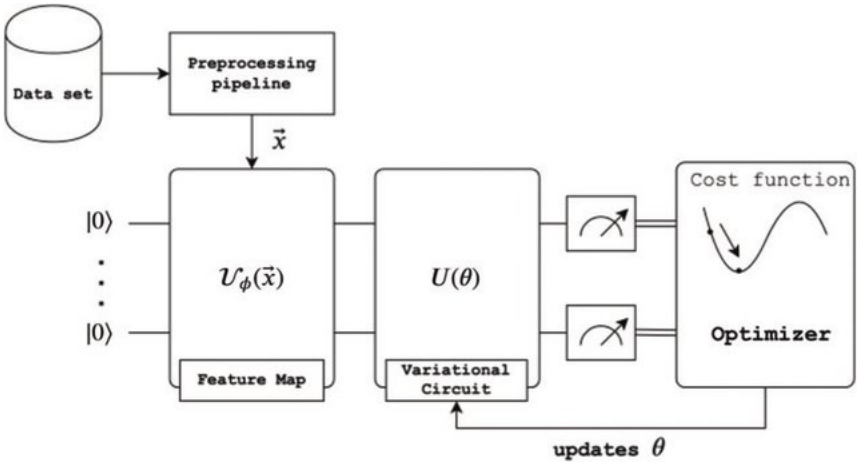


Figure 3.4: Block diagram of Variational Quantum classifier [7]

3.4 Quantum Support Vector Classifier

Classical machine learning has support vector classifier that use kernel trick to classification task. It is not efficient to computing kernel classically for some types of problem, so now we use quantum computing to estimate the kernel using feature map. Taking analogy from SVM, Quantum support vector machine (classifier) in quantum computers has created that is hybrid method using both classical and quantum computers. quantum computers only used to estimate the kernel. We use feature maps to embed our data points and to build the kernel of the svm out of these quantum states [8, 9]. The idea of the quantum kernel is exactly the same as in the classical case. The classical data x encoded into quantum state as $|\phi(x)\rangle$ We take the inner product of these quantum state as $k(\vec{x_i}, \vec{x_j}) = |\langle\Phi(\vec{x_i})|\Phi(\vec{x_j})\rangle|^2$ with the quantum feature maps. This $k(\vec{x_i}, \vec{x_j})$ is the quantum kernel. After calculating the kernel matrix on the quantum computers we can train the quantum svm the same way as classical svm and Then we use dual optimization problem as it is in classical SVM to obtain lagrange multiplier α and decision boundary and maximum margin hyperplane. An example of using a feature map for kernel is in figure 3.5. For the feature maps we use the $U(\Phi(\vec{x}))H^n$. Where H^n is the Hadamard gate applied to each qubit, where n is the number of qubits. Now we need to extract the information about the quantum kernel from the quantum circuit to feed it into the classical SVM algorithm. This is actually a non-trivial task, because we want to measure the overlap of two states $k(\vec{x}, \vec{z}) = |\langle\Phi(\vec{x})|\Phi(\vec{z})\rangle|^2$. The frequency of the

measurement string gives us an estimate of the overlap [6].

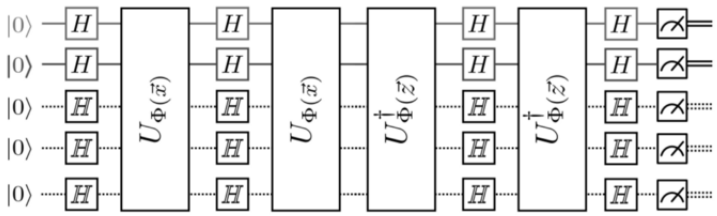


Figure 3.5: Quantum Support Vector Classifier [6]

3.5 CBM Experiment

Compressed Baryonic Matter (CBM) detector is a part of Facility of Antiproton and Ion Research (FAIR) collaboration that operates in the region of transition between hadrons and quark-gluon plasma states. The goal of CBM is to create and characterize super dense nuclei matter in the laboratory. The CBM experimental programme includes a thorough strategy for measuring J/ψ mesons. The produced J/ψ mesons decay into muons, and the CBM detector is utilized in this study. For more details see the CBM physics book [11].

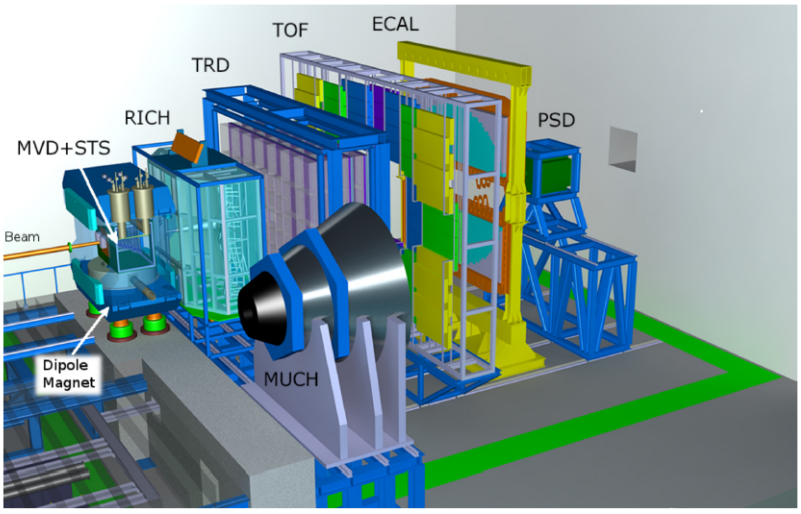


Figure 3.6: CBM Experimental Setup

CBM experimental setup is shown in figure 3.6. After the collision, Silicon Tracking System (STS) is inside the dipole magnet and it is for charged particle

tracking and determine momentum in magnetic field. STS consist of 8 tracking layer of silicon detector at distance of 30 cm to 100 cm from target. After STS , Muon Chamber (MuCh) is placed to identify low momentum muon in environment of high particle densities. hadron absorbers with increasing thickness is placed. There are 12 layers in MuCh. Transition Radiation Detector (TRD) consist 4 layers for particle tracking. TRD is placed Between MuCh and time of flight (TOF) wall. At last the PSD will be used to determine the collision centrality and the orientation of the reaction plane.

In each of the detector, number of layers hit by a track noted down and by path tracking the χ^2 value for particle that is the deviation of particle from its actual position is estimated. There is a cut based method by a linear cut on variables. In this method, if hits are greater than the threshold that is fixed, then the particle is signal or 1 otherwise it will be considered as background or 0. Here machine learning comes into picture to use this method for large amount of data, and various variables are used as machine learning features. Using these values as features, machine learning algorithms predicts weather it is muon rest of the other particles(hadron). The classical Support Vector Machine (SVM) is used for this classification task. In quantum machine learning, Quantum Support Vector Classifier and Variational Quantum Classifier are used. 10 features have been choosen according to their dependency on each other and impact on measurements. They are as : four detectors hits (STS, MuCh, TRD, TOF) and three χ^2 value of detectors (STS, MuCh, TRD) and a one χ^2 value of prime vertex. Rest of the two features are the energy (E) and the momentum (p) of particle.

3.6 Machine Learning Results

Quantum algorithms have been successfully applied to the CBM experiment data. QSVC has been used for classification and got the accuracy and confusion matrix from its output. VQC has also been used with the COBYLA optimizer [10]. There are different types of optimizer and each of them has different method. Pauli feature map with $\alpha = 0.5$, which is an angle and Y-Pauli gate with single repetition were used in both cases. IBM provides backends such as the Statevector simulator, the Aer simulator, and the Qasm simulator. The statevector simulator has been used to simulate our quantum circuit as it is noise-less, and cloud quantum computing has been utilized to run it on real hardware. Ibmq-manila is chosen from among the available backends because it has the most qubits, quantum volume, and clops. This backend supports quantum computing with up to 5 qubits. Creating a simulator on traditional computers is equally challenging, and the number of events is limited to 40000. Figure 3.7 shows the results of QSVC on the statevector simulator, where increasing the number of events increases the amount of training data, and our quantum model learns and predicts more precisely. That is why there is fluctuation with a small number of events, but when enough events have been utilised to train our model, the accuracy score settles around a modest variation.

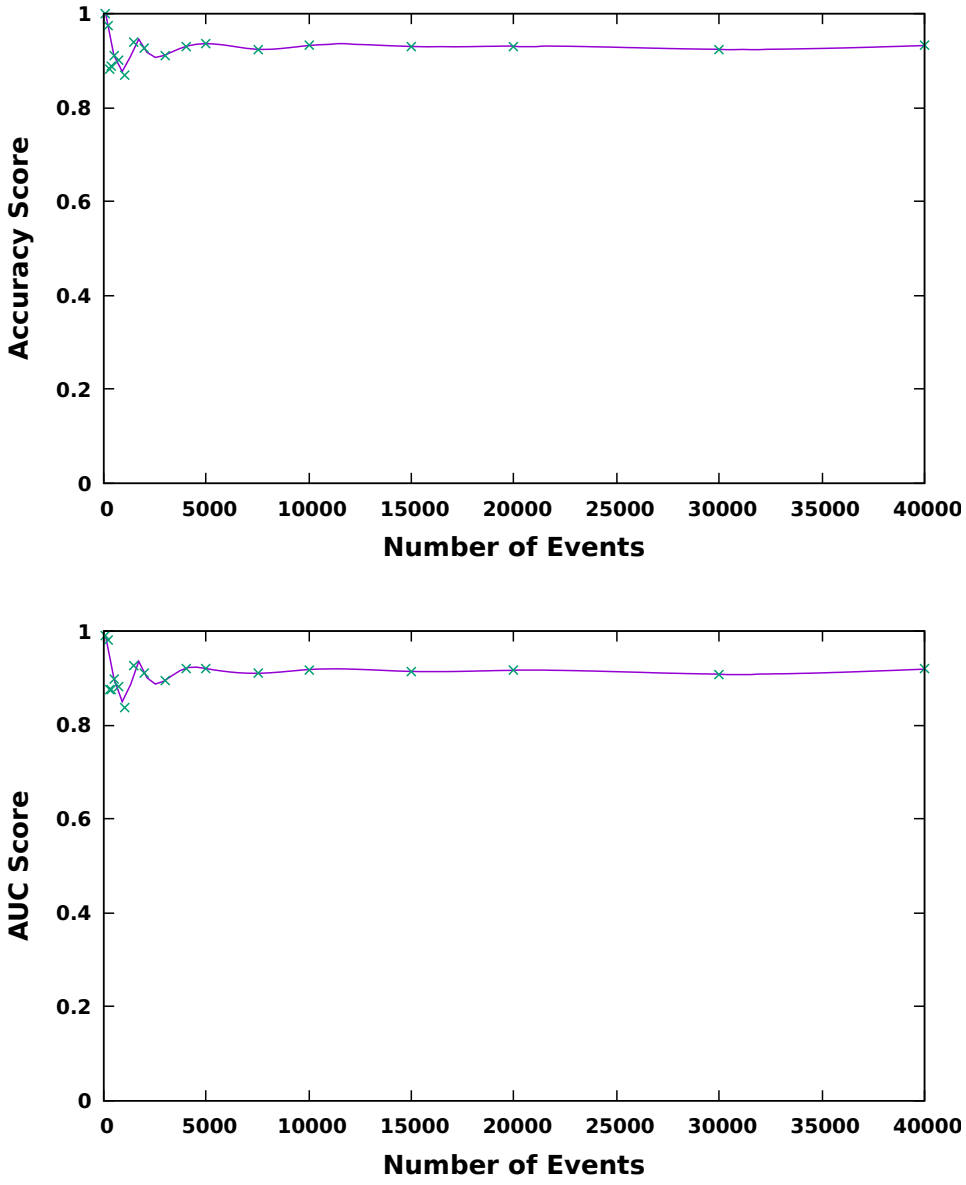


Figure 3.7: Accuracy and AUC score with changing number of events of Quantum Support Vector Classifier on simulator

Figure 3.8 shows the results of VQC on the statevector simulator. With a small number of events, VQC results fluctuate but with a sufficient number of events, it becomes more stable.

Implementing quantum model on real hardware using cloud quantum computing restrict number of feature as it support only 5 qubits. 10 features were used on quantum simulator that discussed in section [3.5]. 5 features that has more impact

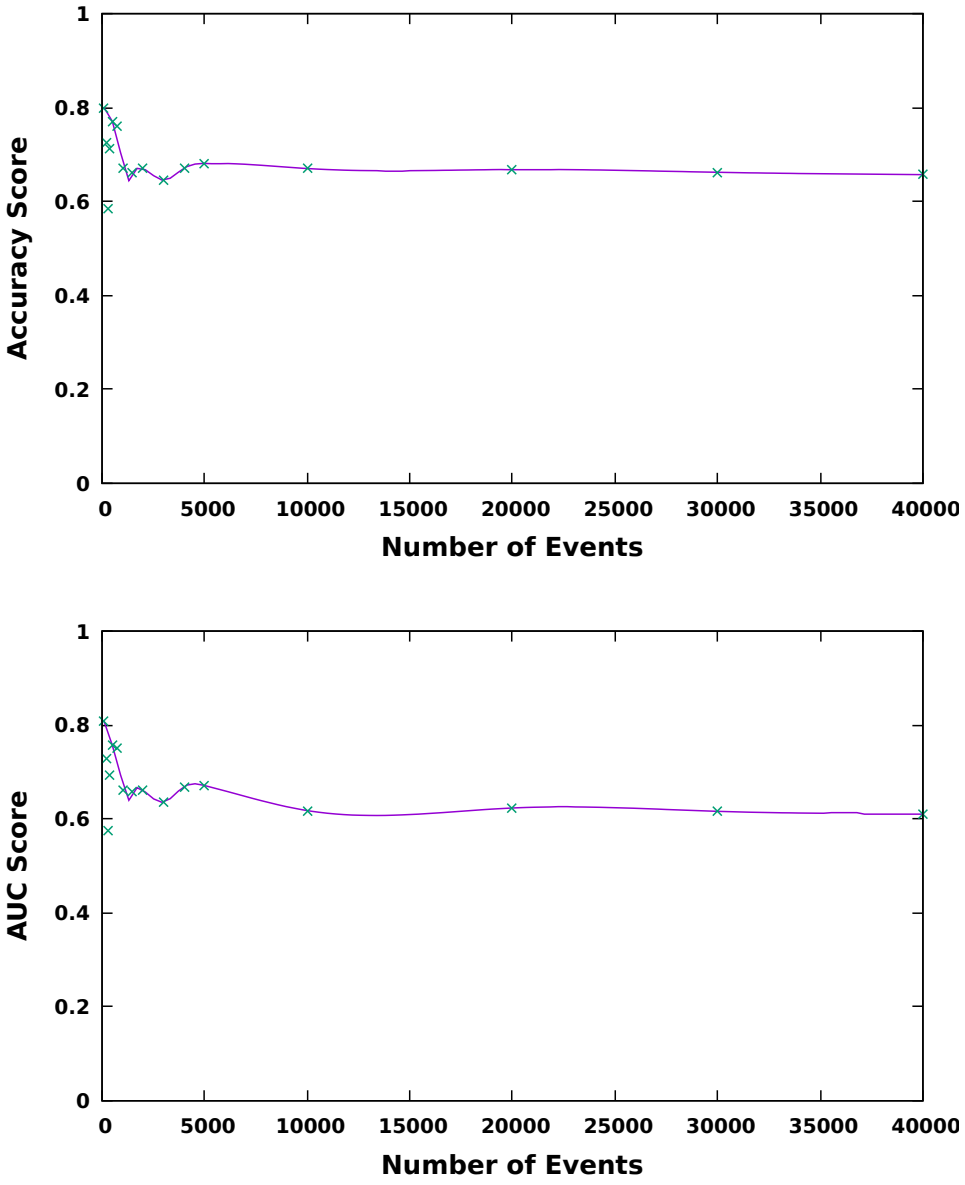


Figure 3.8: Accuracy and AUC score with changing number of events of Variational Quantum Classifier on simulator

on this experiment or they are not related to any other feature have been selected out of 10 features. The selected features are : MuCH hits, STS hits, TRD hits, χ^2-vertex and χ^2-MuCh. For classification, 2000 events were used on Quantum real hardware. With more processing power accessible from quantum computers, more events may be exploited. Quantum machine learning results fluctuate due to the small number of events. When comparing quantum simulator results to real

hardware results, quantum simulation performs better, and the reason for this is that real hardware includes noise, but the simulator we used has no noise. The quantum real hardware results will have an error component, which will affect performance. The results of QSVC running on quantum real hardware are shown

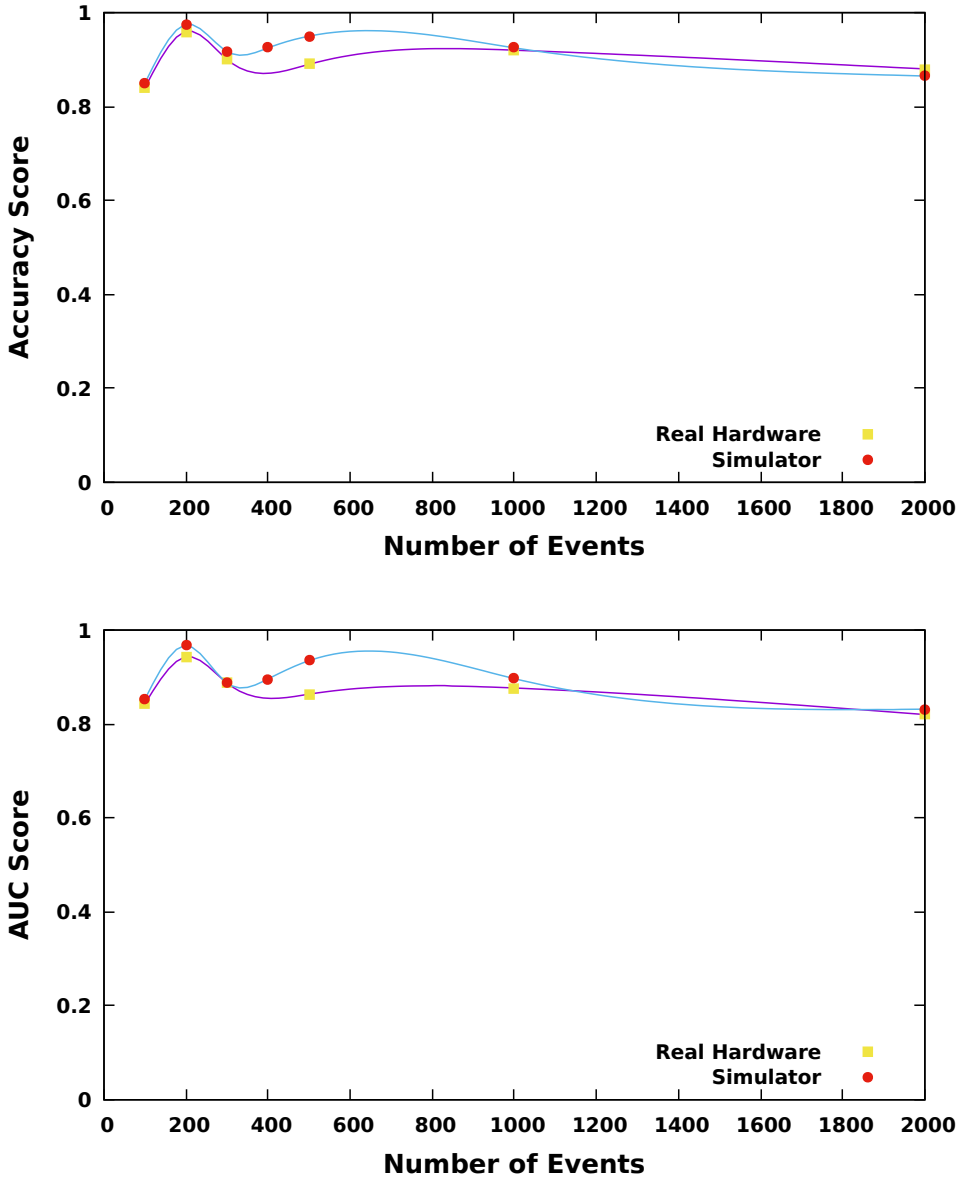


Figure 3.9: Accuracy and AUC score with changing number of events of Quantum Support Vector Classifier on real hardware

in figure 3.9. This shows that the quantum simulator developed is accurate and similar to real hardware with noise. the results of VQC executing on quantum real

hardware are shown in figure 3.10. The results of quantum machine learning were

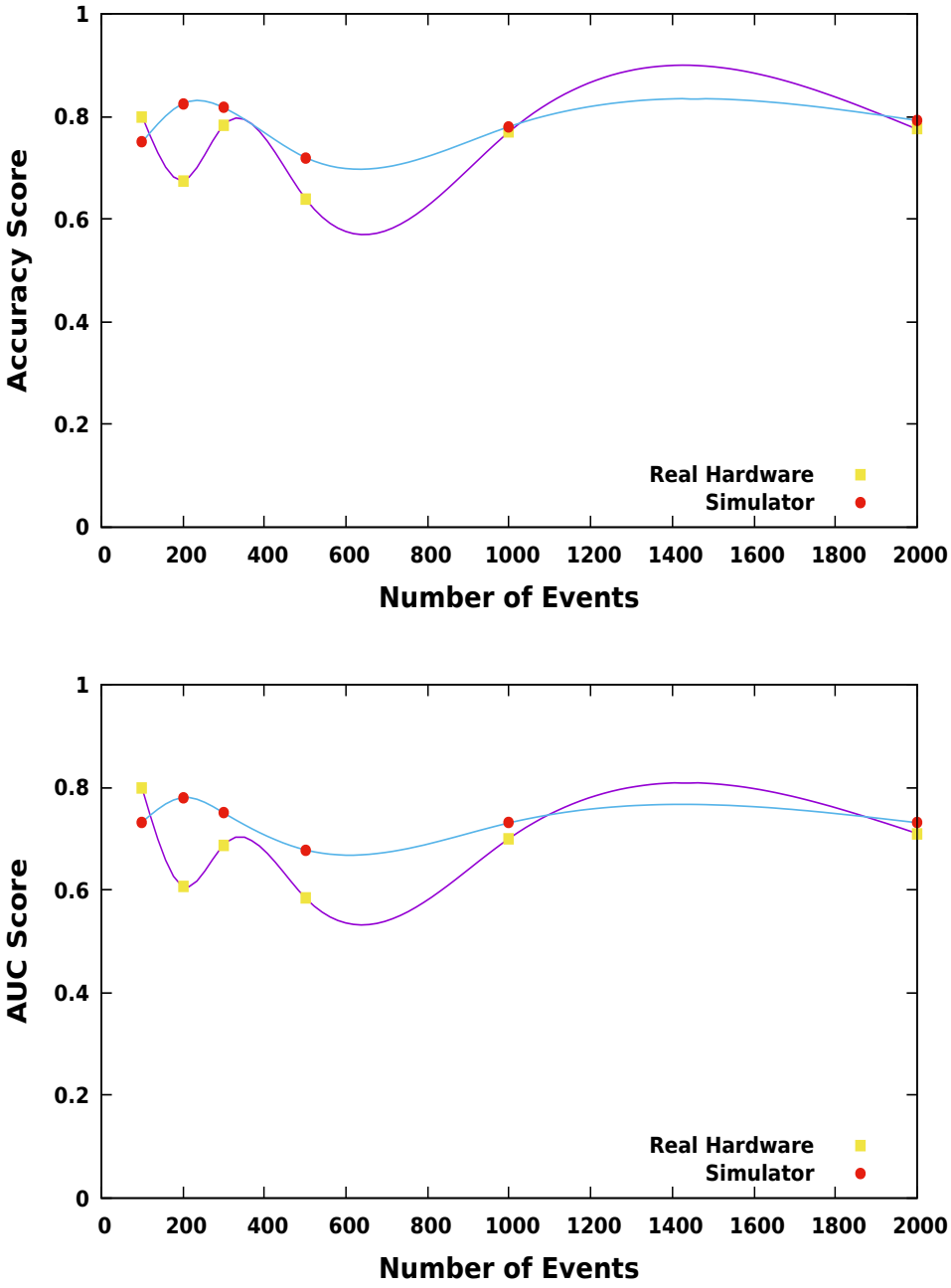


Figure 3.10: Accuracy and AUC score with changing number of events of Variational Quantum Classifier on real hardware

used to compare quantum simulators to real hardware. As SVM has been executed on classical computers with the same features, classical machine learning results

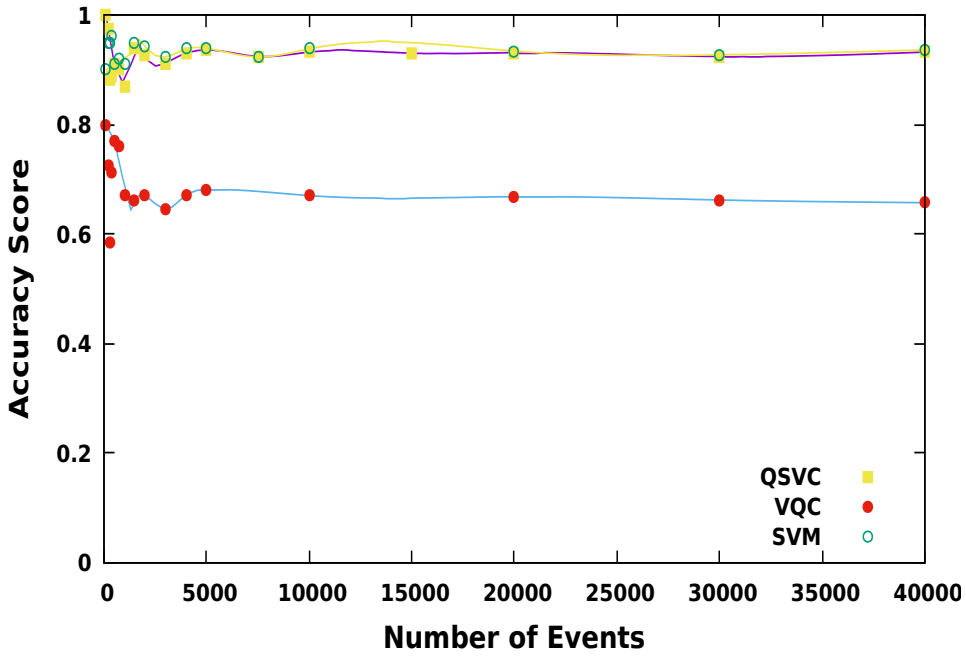


Figure 3.11: Accuracy with changing number of events on SVM, QSVC and VQC

have now been added. SVM is used with a rbf kernel in this instance. Figure 3.11 shows the results of quantum and classical machine learning. When compared to VQC, QSVC produces better results, while classical SVM produces results that are similar to QSVC. VQC's accuracy score is 0.657 or 65.7% when 40000 events are used with 10 features, and it takes around 11 hours to execute on the quantum simulator, whereas QSVC's accuracy score is 0.932 or 93.2% when 40000 events and the same features are used, and it takes around 11 hours to execute on the quantum simulator. Classical SVM with 40000 events and the same 10 characteristics runs in 22 seconds on classical computers and has an accuracy score of 0.936, or 93.6%. The time gap between classical and quantum machine learning is due to the time it takes to create a simulator model on classical processors. In real hardware, based on quantum models that have been developed, data points are splitted into jobs and send it to real hardware. There is a queue to execute this job. We must wait until all of the jobs have been execute successfully.

Chapter 4

Quantum Simulation

Many body system has various types of particle and their interaction terms are also different. Due to many interactions, it is a difficult task to simulate a many body system on classical computers. However classical computers has able to simulate a certain limit of few body system. Here quantum computers may provide a way to simulate these system as it has more processing power. As of an kick start of this field, two body system has simulated on quantum computers and further many body system will simulated. In this thesis, we start with deuteron as two body system and try to simulate it. For quantum simulation, variational quantum eigensolver (VQE) technique has been proposed. This VQE takes input as particle and its hamiltonian, and by using a optimizer it try to get minimum energy which is the ground state energy of this hamiltonian.

4.1 Variational Quantum Eigensolver

Variational Quantum Eigensolver (VQE) is a hybrid algorithm, and as input it needs Hamiltonian of the system. VQE is designed to find the ground state energy of the system by estimating the eigenvalue (energy) of this hamiltonian. After eigenvalue calculation an optimizer is used to get it's lowest value (ground state energy). The Hamiltonian we have is in fermionic operator (creation and annihilation operator) form, to use this Hamiltonian in VQE, it needs to be encoded the hamiltonian into pauli operators form. Why we chooses to get hamiltonian into pauli operator form has simple answer that quantum computers has gates as pauli matrices, so we can easily calculate hamiltonian expectation value using quantum computing. To

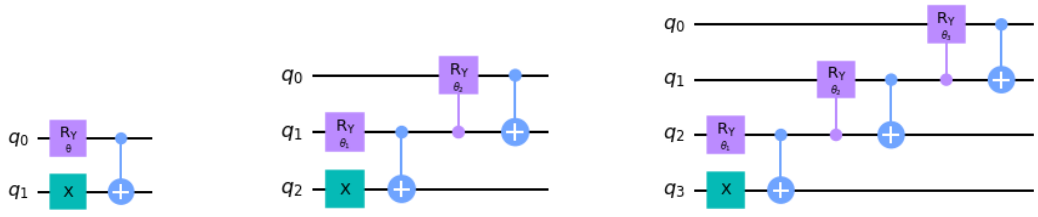
encode these fermionic operators into spin or pauli operators, there are methods for encoding *i.e.,* Gray Code, Bravyi-Kitaev and One-Hot encoding [12]. One-Hot encoding will be used in our work through Jordan-Wigner transformation. After encoding this hamiltonian into pauli operators form or into qubit form, we take an arbitrary state ψ and take expectation value of hamiltonian in this state. In quantum mechanics the expectation value of a operator with respect to a state is always greater then or equal to its lowest eigenvalue and the state should be normalized.

$$a_0 = \langle\psi|\mathrm{H}|\psi\rangle$$

$a_0 \geq a_{min}$, here a_{min} is lowest eigenvalue of operator and a_0 is the expectation value of operator with respect to $|\psi\rangle$ state . The arbitrary state has variational parameter θ, and using classical optimizer we try to get minimum value of this expectation value. This process repeated until we achieved lowest eigenvalue. a_θ is expectation value of parameterized state.

$$a_\theta = \langle\psi(\theta)|\mathrm{H}|\psi(\theta)\rangle$$

The parameterized state $|\psi(\theta)\rangle$ in quantum circuit has been created by applying unitary transformation on initial state $|0\rangle$ as $\mathrm{U}(\theta)|0\rangle = |\psi(\theta)\rangle$, this unitary transformation has combination of quantum gates. Rotational gates around y axis (R_y), controlled not gates, and X gates have been used here and the quantum circuit diagram of state preparation is in figure 4.1 according to number of qubit.



(a) (b) (c)

Figure 4.1: Quantum circuit for variational ansatz for N = 2, 3, 4

The hamiltonian in pauli operator form is used in quantum circuit by applying pauli gates. We take expectation values of all pauli strings in the hamiltonian that is measured. For measurement of pauli strings in hamiltonian, we have to choose

what value of measured quantities we should accept and which one we should not. for single qubit, the expectation value of Z in measurement is taken in arbitrary state $|\psi\rangle$. Since the Z is a form of hamiltonian, so this expectation value is in the form of energy and written as:

$$E_\psi(Z) = \langle\psi|Z|\psi\rangle = \langle\psi|(|0\rangle\langle 0| - |1\rangle\langle 1|)|\psi\rangle$$

$$= \langle\psi|0\rangle\langle 0|\psi\rangle - \langle\psi|1\rangle\langle 1|\psi\rangle$$

$$= |\langle 0|\psi\rangle|^2 - |\langle 1|\psi\rangle|^2$$

$$= P_{|\psi\rangle}(0) - P_{|\psi\rangle}(1)$$

Here $P_{|\psi\rangle}(0)$ is the probability that state $|\psi\rangle$ is measured to be $|0\rangle$, So if 0 measured C_0 times and 1 measured C_1 times out of total measurements. Expectation value of Z is

$$E_\psi(Z) = \frac{C_0 - C_1}{C_0 + C_1}$$

now for expectation value of X

$$E_\psi(X) = \langle\psi|X|\psi\rangle = \langle\psi|HZH|\psi\rangle$$

$$= \langle H^\dagger\psi|Z|H\psi\rangle = \langle H\psi|Z|H\psi\rangle$$

$$= E_{H\psi}(Z)$$

$$= P_{H|\psi\rangle}(0) - P_{H|\psi\rangle}(1)$$

Here $P_{H|\psi\rangle}(0)$ is probability that state $H|\psi\rangle$ is measured to be $|0\rangle$, here we have prepare state $|\psi\rangle$ and apply hadamard gate on it.
For expectation value of Y it can be written as $Y = SXS^\dagger = SHZHS^\dagger$

$$E_\psi(Y) = \langle\psi|SHZHS^\dagger|\psi\rangle$$

$$= \langle H^\dagger S^\dagger\psi|Z|HS^\dagger\psi\rangle$$

$$= \langle HS^\dagger\psi|Z|HS^\dagger\psi\rangle = E_{HS^\dagger|\psi\rangle}(Z)$$

$$= P_{HS^\dagger|\psi\rangle}(0) - P_{HS^\dagger|\psi\rangle}(0)$$

Here we have to prepare a state $|\psi\rangle$ and apply hadamard and $S^\dagger$ gate. For 2 qubit

XX expectation value is

$$E_\psi(XX) = \langle\psi|XX|\psi\rangle = \langle\psi|(HZH)\otimes(HZH)|\psi\rangle$$

$$= \langle H^\dagger H^\dagger\psi|Z\otimes Z|HH\psi\rangle$$

$$= \langle HH\psi\,|(|0\rangle\langle 0| - |1\rangle\langle 1|)\otimes(|0\rangle\langle 0| - |1\rangle\langle 1|)|\,HH\psi\rangle$$

$$= \langle HH\psi|(|00\rangle\langle 00| - |01\rangle\langle 01| - |10\rangle\langle 10| + |11\rangle\langle 11|)|HH\psi\rangle$$

$$= P_{HH|\psi\rangle}(00) - P_{HH|\psi\rangle}(01) - P_{HH|\psi\rangle}(10) + P_{HH|\psi\rangle}(11)$$

Other combination of XY and ZZ can be calculated same way and for more number of qubit we can use this calculation and choose the results that we have to take. This results are used by classical optimizer to choose a set of parameters for which eigenvalue is minimized. for better optimization, the state should be close to ground state. This parameterized state is also called variational ansatz and it depends on encoding and corresponding transformation.

4.2 Deuteron Problem

Deuteron as two body system consist a proton and a neutron. Ground state energy calculation of deuteron will establish the concept of quantum simulation and it can be used to simulate heavier nuclei. For VQE input, we use a Hamiltonian from pionless effective field theory [15, 16] and have to work on discrete variable representation. For more detail one can look appendix A. Hamiltonian for deuteron is,

$$\mathrm{H}_N = \sum_{n,n'=0}^{N-1} \langle n'|\mathrm{T}+\mathrm{V}|n\rangle a^\dagger_{n'} a_n$$

Kinetic energy matrix element and potential energy matrix element from [14] are

$$\langle n'|\mathrm{T}|n\rangle = \frac{\hbar\omega}{2}\left[\left(2n+\frac{3}{2}\right)\delta^{n'}_n - \sqrt{n\left(n+\frac{1}{2}\right)}\delta^{n'+1}_n - \sqrt{(n+1)\left(n+\frac{3}{2}\right)}\delta^{n'-1}_n\right],$$

$$\langle n'|\mathrm{V}|n\rangle = V_0\delta^0_n\delta^{n'}_n.$$

Here, $V_0 = -5.68658111$ MeV, $a^\dagger_{n'}$ and a_n is creation and annihilation operator and n′ and n are state of harmonic oscillator and N is dimension of oscillator. Values

of T and V can be directly use from above equations of energy matrix elements. This harmonic oscillator hamiltonian operator is fermionic as is consists fermionic creation and annihilation operator. In second quantization form the hamiltonian is

$$\mathrm{H} = \sum_{ij} h_{ij} a_i^\dagger a_j$$

here this h_{ij} is :

$$h_{ij} = \langle i|T + V|j\rangle = \int \phi_i(r) \left[\frac{\hbar^2}{2m}\nabla^2 + V(r)\right] \phi_j(r) dr$$

here $\phi_i(r)$ or $|i\rangle$ represent the basis of system. So we calculate h_{ij} manually and $a^\dagger a$ will be evaluated by quantum computing. But $a^\dagger a$ can not be directly calculate by quantum computing. Basis state and operator has to be mapped into qubit's basis state and qubit operations. Qubit basis states are $|0\rangle$ and $|1\rangle$ and a state can be represented as $\psi_q = a|0\rangle + b|1\rangle$ and for n-qubit system

$$\psi_q^n = \otimes \prod_{j=0}^{n-1} (a_j|0\rangle + b_j|1\rangle)$$

For quantum computing we have to map ψ_f into ψ_q and mapping decide some factor of computing as number of qubits, number of operations and circuit depth. In JW transformation a single qubit will be assign of one basis state $|i\rangle$.
The quantum spin with S = 1/2 can describe by fermions as $|\downarrow\rangle = f^\dagger|0\rangle$ and $|\uparrow\rangle = |0\rangle$. Spin up and down states of single spin corresponds to empty fermion states and single occupied state. We can take the analogy from fermionic operator and spin-1/2 Pauli operators can transform as $\sigma^+ \equiv f$ and $\sigma^- \equiv f^\dagger$ [17]. The spin-1/2 Pauli operator is σ^x, σ^y and σ^z and representation of spin lowering and raising operator is

$$\sigma^+ = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

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

σ_j defined as Pauli spin operator acting on state j or site j, spin raising and lowering

operator σ_j^+ and σ_j^- in terms of x and y component of spin operator is

$$\sigma_j^+ = (\sigma_j^x + \iota\sigma_j^y)/2$$

$$\sigma_j^- = (\sigma_j^x - \iota\sigma_j^y)/2$$

The anti-commutator relation of σ_j^+ and σ_j^- is $\{\sigma_j^+, \sigma_j^-\} = 1$, which is expected as from fermionic operator follows the same relation for same site, but on different site fermionic operator has $[f_j^\dagger, f_k] = 0$ for $j \neq k$ and spins are unlike the fermions on different site anti-commutation. To have the analogy from fermions, we have to get fermion commutation relations from spin operators. This can be done by taking a new set of operators which is a transformation of fermionic operator defined as :

$$a_j^\dagger = e^{\left(+\iota\pi \sum_{k=1}^{j-1} f_k^\dagger f_k\right)}.f_j^\dagger$$

$$a_j = e^{\left(-\iota\pi \sum_{k=1}^{j-1} f_k^\dagger f_k\right)}.f_j$$

$$a_j^\dagger a_j = f_j^\dagger f_j$$

We just added a phase factor called string which is determined by number of occupied fermionic modes in $k = 1, ..., j-1$ and phase factor can be written as

$$e^{\left(+\iota\pi \sum_{k=1}^{j-1} f_k^\dagger f_k\right)} = \prod_{k=1}^{j-1} e^{+\iota\pi f_k^\dagger f_k} = \prod_{k=1}^{j-1}(1 - 2f_k^\dagger f_k) = \prod_{k=1}^{j-1}(\sigma_k^z)$$

Here $\sigma_k^z = 1 - 2f_k^\dagger f_k$. The canonical anti-commutation relation is $\{a_j^\dagger, a_j\} = 1$, $\{a_j^\dagger, a_k^\dagger\} = 0$ and $\{a_j, a_k\} = 0$ and followed by spin operator, So now we can take the analogy

$$\sigma_j^+ = e^{\left(-\iota\pi \sum_{k=1}^{j-1} f_k^\dagger f_k\right)}.a_j$$

$$\sigma_j^- = e^{\left(+\iota\pi \sum_{k=1}^{j-1} f_k^\dagger f_k\right)}.a_j^\dagger$$

$$\sigma_j^z = 1 - 2a_j^\dagger a_j$$

Here $a_j^\dagger a_j = f_j^\dagger f_j$ is number operator which can have value of 0 and 1. Now to map fermionic operator in spin operator we use these above transformation as from σ^z,

$$a_j^\dagger a_j = \frac{(1 - \sigma_j^z)}{2} = \frac{(I - Z_j)}{2}$$

We will use the notations σ_j^x, σ_j^y and σ_j^z as X_j, Y_j and Z_j respectively which represent the X, Y and Z quantum gate on j-qubit.

$$a_j^\dagger a_k = \frac{1}{4}(\sigma_j^x - \iota\sigma_j^y)(\sigma_k^x + \iota\sigma_k^y)$$

$$a_j a_k^\dagger = \frac{1}{4}(\sigma_j^x + \iota\sigma_j^y)(\sigma_k^x - \iota\sigma_k^y)$$

adding these two terms

$$a_j^\dagger a_k + a_j a_k^\dagger = \frac{1}{2}(\sigma_j^x\sigma_k^x + \sigma_j^y\sigma_k^y) = \frac{1}{2}(X_jX_k + Y_jY_k)$$

we can simply write that the Jordan-Wigner transformation of fermionic operators to quantum computing as :

$$a_j^\dagger \to \frac{1}{2}\left[\prod_{k=0}^{j-1} Z_k\right](X_j - \iota Y_j)$$

$$a_j \to \frac{1}{2}\left[\prod_{k=0}^{j-1} Z_k\right](X_j + \iota Y_j)$$

In our fermionic hamiltonian there are two types of terms according to operators and we can transform it into spin operator suitable for quantum computing by

$$h_{jj}a_j^\dagger a_j = \sum_j \frac{h_{jj}}{2}(1 - Z_j)$$

$$h_{jk}(a_j^\dagger a_k + a_k^\dagger a_j) = \frac{h_{jk}}{2}\left(\prod_{p=k+1}^{j-1} Z_p\right)(X_jX_k + Y_jY_k)$$

where h_{jj} and h_{jk} are from fermionic hamiltonian mention above.
for N=1, We calculate spin operator Hamiltonian H_1 :

$$\mathrm{H}_1 = \langle 0|\mathrm{T} + \mathrm{V}|0\rangle a_0^\dagger a_0$$

$$= (\langle 0|\mathrm{T}|0\rangle + \langle 0|\mathrm{V}|0\rangle)a_0^\dagger a_0$$

$$= \frac{\langle 0|\mathrm{T}|0\rangle}{2}(I - Z_0) + \frac{\langle 0|\mathrm{V}|0\rangle}{2}(I - Z_0)$$

taking value of $\langle 0|\mathrm{T}|0\rangle$ and $\langle 0|V|0\rangle$ from kinetic and potential energy matrix

$$\mathrm{H}_1 = \frac{3\hbar\omega}{8}(I - Z_0) + \frac{V_0}{2}(I - Z_0)$$

taking the value of $\hbar\omega = 7$ MeV, the reason for taking this value will be discussed in next section on variational calculation.

$$\mathrm{H}_1 = 2.625(I - Z_0) - 2.48432(I - Z_0) = 0.218291(Z_0 - I)$$

For N=2

$$\begin{aligned}
\mathrm{H}_2 &= \langle 0|T + V|0\rangle a_0^\dagger a_0 + \langle 0|T + V|1\rangle a_0^\dagger a_1 + \langle 1|T + V|0\rangle a_1^\dagger a_0 + \langle 1|T + V|1\rangle a_1^\dagger a_1 \\
&= \langle 0|T|0\rangle a_0^\dagger a_0 + \langle 0|V|0\rangle a_0^\dagger a_0 + \langle 0|T|1\rangle a_0^\dagger a_1 + \langle 1|T|0\rangle a_1^\dagger a_0 + \langle 1|T|1\rangle a_1^\dagger a_1 \\
&= \frac{3\hbar\omega}{8}(I - Z_0) + \frac{V_0}{2}(I - Z_0) - \frac{\hbar\omega}{4}\sqrt{\frac{3}{2}}(X_0X_1 + Y_0Y_1) + \frac{7\hbar\omega}{8}(I - Z_1) \\
\mathrm{H}_2 &= 5.9067I + 0.218291Z_0 - 6.125Z_1 - 2.143304(X_0X_1 + Y_0Y_1)
\end{aligned}$$

From similar way we can transform for N=3,4, and so on and value of

$$\begin{aligned}
\mathrm{H}_3 = {} & 15.531709I + 0.218291Z_0 - 6.125Z_1 - 9.625Z_2 - 2.143304(X_0X_1 + Y_0Y_1) \\
& - 3.9133119(X_1X_2 + Y_1Y_2)
\end{aligned}$$

$$\begin{aligned}
\mathrm{H}_4 = {} & 28.656709I + 0.218291Z_0 - 6.125Z_1 - 9.625Z_2 - 13.125Z_3 \\
& - 2.143304(X_0X_1 + Y_0Y_1) - 3.9133119(X_1X_2 + Y_1Y_2) - 5.670648(X_2X_3 + Y_2Y_3)
\end{aligned}$$

Now we can use this hamiltonian in VQE and can estimate the expectation value and optimize it for ground state energy. Results from VQE is in table 4.1 .

H_N	Energy
H_1	-0.436 MeV
H_2	-1.749 MeV
H_3	-2.045 MeV
H_4	-2.143 MeV

Table 4.1: Ground state energy of deuteron for increasing the dimension of harmonic oscillator

Since this is for only N=4, for higher number of N our results are more better, but we have limitation in number of qubit. Extrapolation technique is used here for this problem, however if more number of qubits is available then results would be more precise without extrapolating. In next section we talk about extrapolation and required parameters for it.

4.3 Variational Calculation in Harmonic Oscillator Basis

The harmonic oscillator basis offers an expansion basis that is widely used in nuclear structure computations, but due to restricted computational resources, the basis must be truncated before computation. An extrapolation result to an infinite basis size is required in this computation. Truncating a harmonic oscillator basis has two variables, dimension of harmonic oscillator basis N and harmonic oscillator energy parameter $\hbar\omega$. Variational calculation utilize to estimate these variables, so we can define the size of expansion basis or model space. These two variables are associated with two momentum cutoff known as Ultraviolet momentum cutoff and Infrared momentum cutoff. Ultraviolet momentum cutoff related to the energy of the highest harmonic oscillator level in harmonic oscillators, whereas infrared cutoff (λ) corresponds to the lowest allowable momentum difference between single particle orbitals.

The truncated space defined by N and $\hbar\omega$ can now be considered a model space characterised by two momentum. The Ultraviolet momentum

$$\Lambda = \sqrt{m_N(N + 3/2)\hbar\omega}$$

here m_N is nucleon mass and taken $m_N = 938.2$ MeV. To obtain this expression of UV momentum, establish kinetic energy to one-half of total energy at the highest harmonic oscillator level, then solve the non-relativistic dispersion relation for Λ. Energy levels of harmonic oscillator is quantized in units of $\hbar\omega$ and momentum difference is taken as infrared cutoff that is low momentum cutoff

$$\lambda = \sqrt{m_N \hbar\omega} = \hbar/b$$

here b = $\sqrt{\hbar/m_N\omega}$ postulated as oscillator length. Influence of the infrared cutoff is removed by extrapolating to the continuum limit, where $\hbar\omega \to 0$ with $N \to \infty$ so that Λ is fixed. We can not achieve both the ultraviolate limit and infrared limit by taking $\hbar\omega$ to zero in a fixed N model space as this procedure take ultraviolate cutoff to zero.

Another definition of infrared momentum is the maximal radial extent needed to envelop the system we attempt to describe by finite basis space. it is defined as

$$\lambda_{sc} = \sqrt{(m_N \hbar\omega)/(N + 3/2)}$$

To distinguish between two notation, λ stands for first definition and λ_{sc} stands for second definition because of its scaling properties. When oscillator length b is large enough (frequency is small enough) the ultraviolate correction are negligible in compared to infrared corrections with fixed Λ. Controlled extrapolation to the results for the full model space can be made if UV and IR corrections can be understood formally.

We fix one cutoff and vary the other one; the ultraviolet cutoff value Λ is fixed in this scenario, whereas the infrared cutoff value λ_{sc} is varied. As from [20], the IR regulator is a function of λ_{sc} and not λ. The IR cutoff, as previously stated, is the lower cutoff, and we try to change it for low values. Figure 4.2 shows that for larger value of Λ, $|\Delta E/E|$ converges to zero and if we decrease the value of λ_{sc}, $|\Delta E/E|$ starts converging towards zero. We can see the changes for $\Lambda = 140$ MeV/c and $\Lambda = 150$ MeV/c. This suggests that there should be some cutoff for UV momentum, below that $|\Delta E/E|$ dose not converge towards zero. We can describe it as $\Lambda \geq \Lambda^{NN}$ where Λ^{NN} is ultraviolate regulator scale of NN interaction. From the figure 4.2 we can say that $\Lambda^{NN} \sim 150$ MeV/c. This suggest that λ_{sc} could be

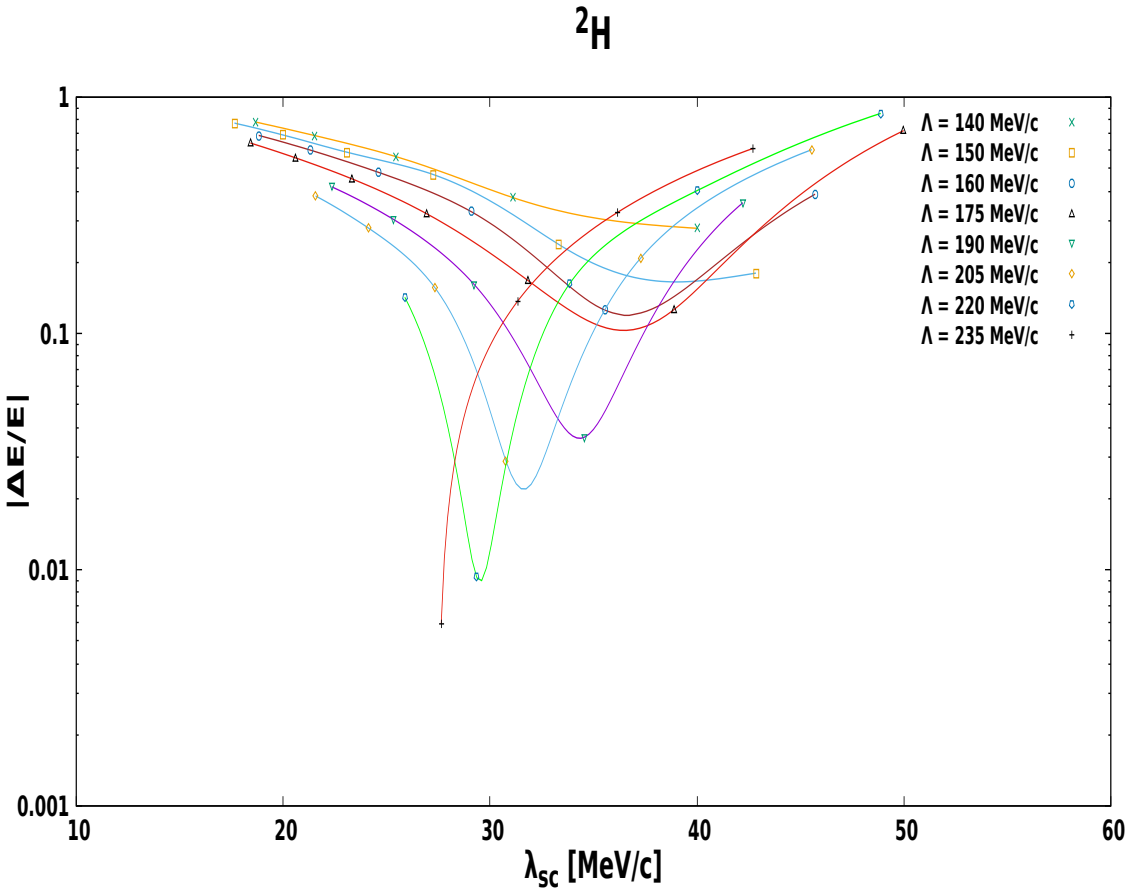


Figure 4.2: Dependency of ground state energy of deuteron upon Infrared cutoff (λ_{sc}) for fixed Λ

used for extrapolation to the IR limit ($\lambda_{sc} \to 0$) provided that Λ is large enough to capture the UV region of the calculation by taking $\Lambda \geq \Lambda^{NN}$.

Lowering the IR cutoff will include more of the IR region in the calculation, thus $|\Delta E/E|$ does not go to zero for smaller Λ. Now we fix the IR cutoff for running $|\Delta E/E|$ upon the UV cutoff Λ. As in figure 4.3 at small value of Λ,$|\Delta E/E|$ does not converged to zero but increasing the value of Λ will start converging it to zero. From the figure 4.3 the curves above $\lambda_{sc} = 38$ MeV/c has high λ_{sc} value and the curves is not converging above this value. λ_{sc} have some cutoff value, below this our results will converge $\lambda_{sc} \leq \lambda_{sc}^{NN}$. Here λ_{sc}^{NN} is second characteristic IR regulator scale. For increasing the value of Λ, our results for $|\Delta E/E|$ will not close to zero this indicates that we have more of UV physics and when Λ is low and λ_{sc} is also below the cutoff value then it has more of IR physics. From the figure 4.4, the line coming

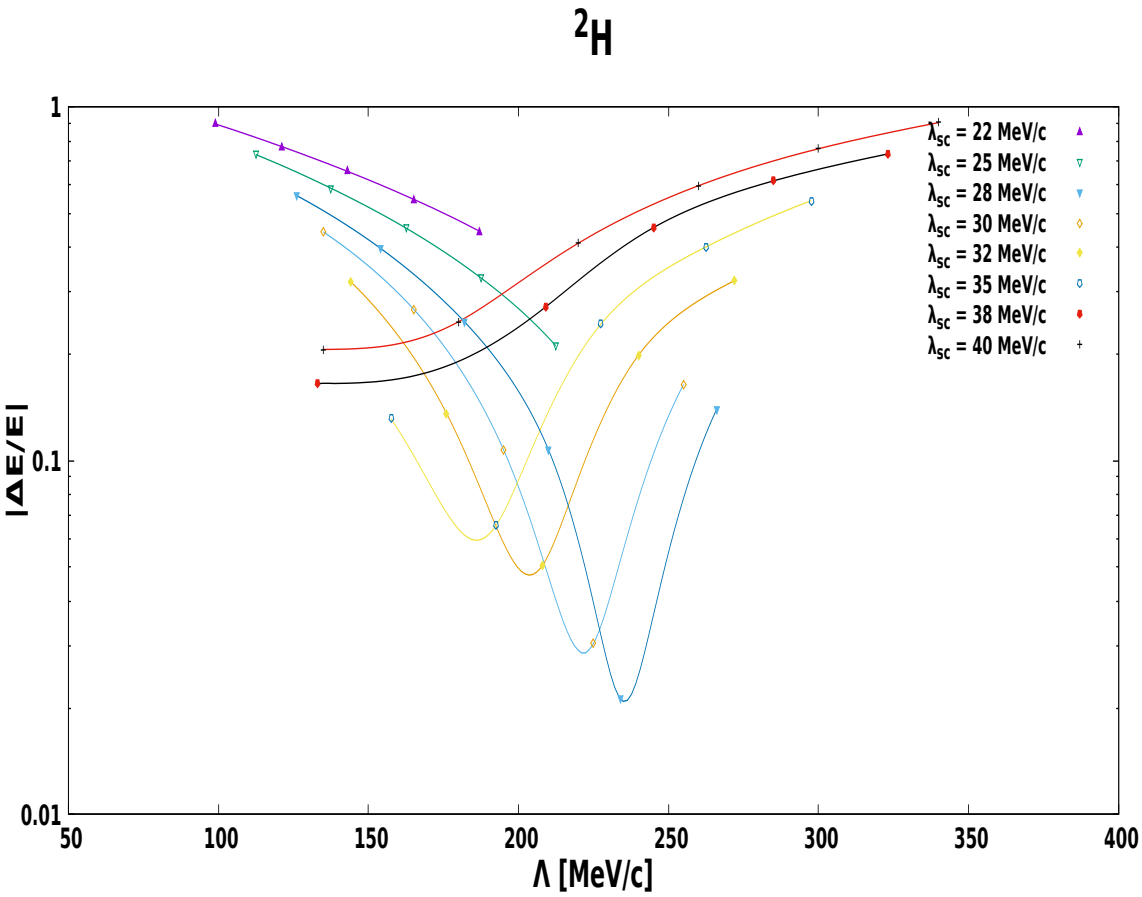


Figure 4.3: Dependency of ground state energy of deuteron upon infrared momentum cutoff (Λ) for fixed λ_{sc}

out from the rest of the curves has the λ_{sc} value at the cutoff. So by increasing the Λ we just add more of the uv physics and IR physics is already there. so on these value we have the best extrapolated results and now we can apply leading order extrapolation formula.

The first definition of infrared momentum cutoff $\lambda = \sqrt{m_N \hbar\omega}$ has to be fit in such a way that the $|\Delta E/E|$ tends to zero. Then by varying the λ value close to zero, we try to get best fit value of $\hbar\omega$ for increasing value of Λ. As it can be seen from figure 4.5 that for increasing value of Λ, our curves start to converge on $\lambda = 81$ MeV/c. This indicates that the value for $\hbar\omega$ is 7 MeV and on this particular value, increasing the UV cutoff Λ, will tend our $|\Delta E/E|$ to zero. In this way, we have used both UV and IR region of correction. Lowering the IR cutoff and raising the

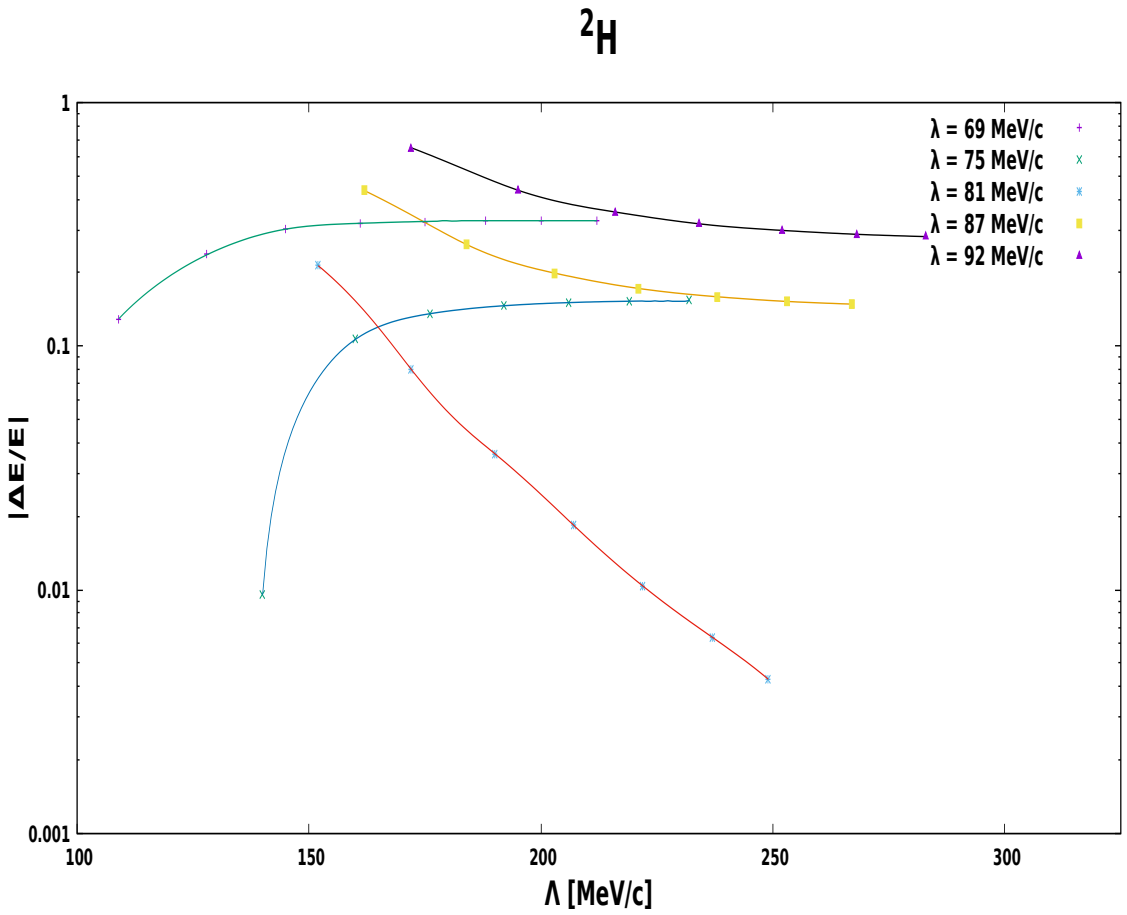


Figure 4.4: Dependency of ground state energy of deuteron upon Ultraviolet cutoff (Λ) for fixed λ

UV cutoff, leads to more exact results, however $\hbar\omega$ also lowers the IR cutoff, thus losing UV physics and boosting IR physics leads to worse extrapolation results. As a consequence, we determine the $\hbar\omega = 7$ MeV, on which increasing the UV momentum yields the optimal extrapolated results.

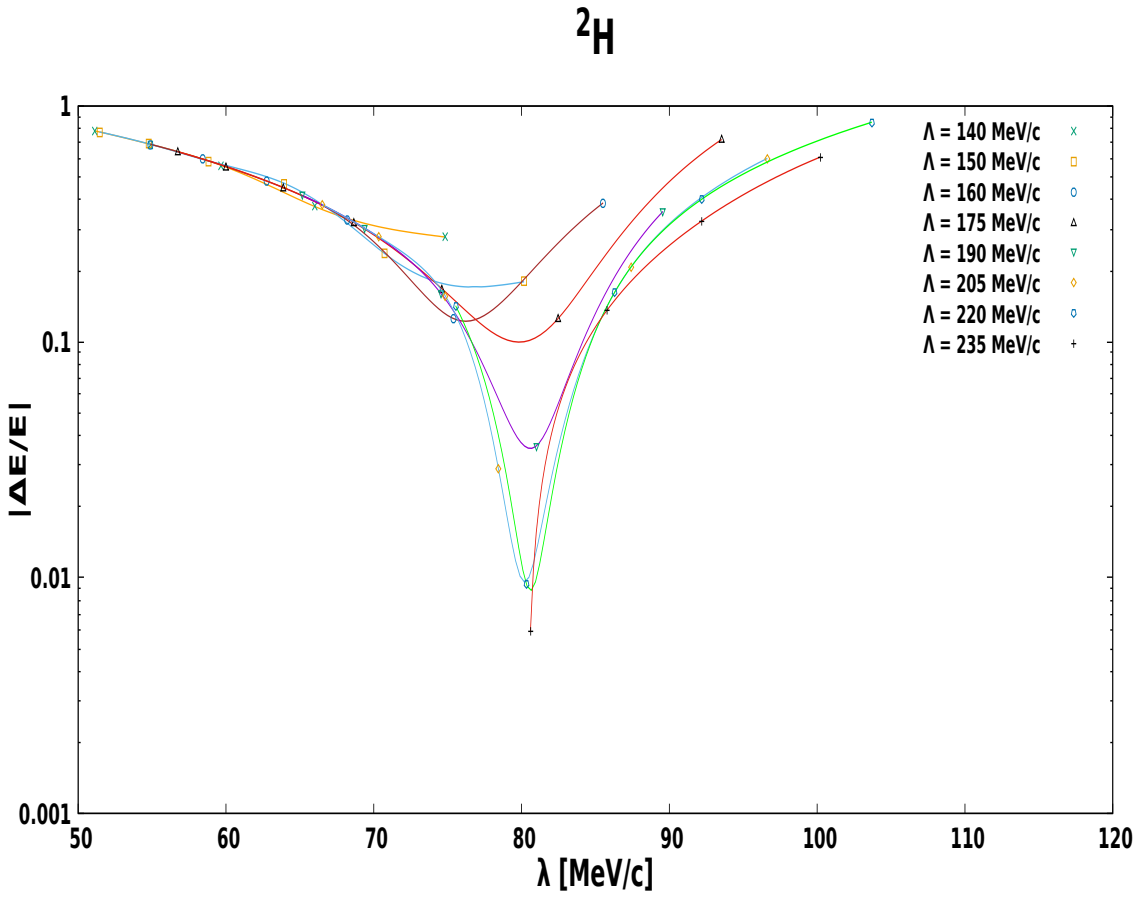


Figure 4.5: Dependency of ground state energy of deuteron upon infrared momentum cutoff (λ) for fixed Λ

4.4 Quantum Simulation Results

As discussed in variational calculation that we have to choose a fit value for IR cutoff and then can increase the value of UV cutoff to have best results in extrapolation. For deuteron problem $\hbar\omega = 7$ MeV has taken and use leading order extrapolation, here Luscher formula [18] for extrapolation can be used.

$$E(L) = E_\infty + Ae^{-2k_\infty L} + \mathcal{O}(e^{-4k_\infty L})$$

This is L dependent which is effective Dirichlet boundary condition [18] and $k_\infty = \sqrt{-2mE_\infty/\hbar^2}$ is binding momentum defined from separation energy E_∞. Harmonic oscillator variant of Luscher formula for finite size correction is

$$E_N = -\frac{\hbar^2 k^2}{2m}\left(1 - 2\frac{\gamma^2}{k}e^{-2kL} - 4\frac{\gamma^4 L}{k}e^{-4kL}\right)$$

Here finite basis energy E_N in term of infinity basis energy E_∞ and small extrapolation correction terms. Finite basis energy has been calculated by computing and using above formula, infinity basis energy estimated. Here k is bound state momentum and γ is asymptotic normalization coefficient and N is dimension of oscillator basis. For N = 2, $\Lambda = 150$ MeV/c while $\hbar\omega = 7$ MeV. quantum computing has been used to calculate ground state energy for N = 2 and estimate value for $E_N = -1.749$ MeV, this allows one to fit correction by adjusting k and γ [14]. After the correction we get the extrapolated ground state energy for N = 2 is -2.20 MeV which is about 1% away from deuteron's ground state energy of -2.224 MeV

Chapter 5

Conclusion

We have successfully applied the quantum machine learning on high energy physics data and the results from quantum machine learning are satisfactory. If we compare the results with classical machine learning results then QSVC is able to achieve the bench marked set value of SVM. Results from VQE is not that good as compare to QSVC. This is an early stage of quantum computing and we have showed a way to use it in high energy physics, so when we have a near-term quantum computer device, we can implement quantum algorithms on it. The complexity of data and amount is increasing day by day and quantum computers has ability to handle it. In some of the work it has shown that it perform better than classical on complex data.

Simulating an atom using quantum computers is not a novelty work but this is the first step towards the quantum simulation where we can use all the physics. As the potential and kinetic terms in hamiltonian is increasing with the size and number of nuclei, it becomes computationally expensive task, here quantum computers comes into picture. As we have more processing power in compare to classical computers. At this time we are bound with limited number of qubit processor but in next few years we will be able to use quantum computers for non-trivial problems that classical can not do and all this work will help the mankind to explore new things.

Appendices

Appendix A

Deuteron Problem

As nucleons are not fundamental particles but a composition of quarks and gluons. So, to understand its complicated structure formed by strong interaction between sub-particles and its dynamics, we study Quantum Chromodynamics (QCD). It's not easy to compute nuclear observables from QCD as the nucleus contains multiple complex states of quarks and gluons that interact with each other. The problem can be solved by Lattice QCD [21] computational technique if nuclear interaction among nucleons is expressed in terms of quarks and gluons. It's computationally expensive since the problem is highly non-perturbative because of strong coupling between quarks and gluons. As a result, we try to connect lattice QCD data to heavier nuclei using low energy effective field theory. In nuclear physics, effective field theory makes use of the hierarchy of energy scales. So, interaction between quarks and gluons are governed by QCD at nucleon mass(1 GeV) or heavy mesons mass ($\sim$ 800 MeV). When we go for low energy our resolution scale is decreases and for nucleon interaction it is done at lightest meson (pion) mass $\approx$ 140 MeV. Deuteron is in ${}^3S_1$ partial wave state and its scattering length in triplet S channel is ${}^{np}a_t = 5.42$ fm [22] and corresponding momentum scale is $1/{}^{np}a_t \sim 36$ MeV. Scattering length is greater then pions physical range of potential (1.4 fm). So for this interaction we have to lower the energy and integrated out the pions from interaction to be in scattering length. When our energy scale is below pions mass, it is called pionless effective field theory. The reason for disparity of S-wave scattering length and QCD scale is not yet understood. We take advantages of separation of scale in pionless EFT as nucleons are slow to resolve pions, we can describe low energy NN potential. Our scale is in range of 30-50 MeV.

Bibliography

[1] Feynman, Richard P. “Simulating physics with computers.” Int. j. Theor. phys 21.6/7 (1982).

[2] Chow, Jerry, Oliver Dial, and Jay Gambetta. “IBM Quantum breaks the 100-qubit processor barrier.” IBM Research Blog (2021).

[3] Nielsen, Michael A., and Isaac Chuang. “Quantum computation and quantum information.” (2002): 558-559.

[4] Schuld, Maria, and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Berlin: Springer, 2018.

[5] Sahin, Mehmet Özgür, Dirk Krücker, and I-A. Melzer-Pellmann. “Performance and optimization of support vector machines in high-energy physics classification problems.” Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 838 (2016): 137-146.

[6] Havlíček, Vojtěch, et al. “Supervised learning with quantum-enhanced feature spaces.” Nature 567.7747 (2019): 209-212.

[7] Sierra-Sosa, Daniel, et al. “Dementia prediction applying variational quantum classifier.” arXiv preprint arXiv:2007.08653 (2020).

[8] Wu, Sau Lan, et al. “Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits.” Journal of Physics G: Nuclear and Particle Physics 48.12 (2021): 125003.

[9] Chan, Jay, et al. "SISSA: Application of Quantum Machine Learning to High Energy Physics Analysis at LHC using IBM Quantum Computer Simulators and IBM Quantum Computer Hardware." PoS (2021): 930.

[10] Joshi, Nisheeth, Pragya Katyayan, and Syed Afroz Ahmed. "Evaluating the Performance of Some Local Optimizers for Variational Quantum Classifiers." Journal of Physics: Conference Series. Vol. 1817. No. 1. IOP Publishing, 2021.

[11] Friman, Bengt, et al., eds. The CBM physics book: Compressed baryonic matter in laboratory experiments. Vol. 814. Springer, 2011.

[12] Siwach, Pooja, and Paramasivan Arumugam. "Quantum simulation of nuclear Hamiltonian with a generalized transformation for Gray code encoding." Physical Review C 104.3 (2021): 034301.

[13] Stetcu, I., et al. "Effective theory for trapped few-fermion systems." Physical Review A 76.6 (2007): 063613.

[14] Dumitrescu, Eugene F., et al. "Cloud quantum computing of an atomic nucleus." Physical review letters 120.21 (2018): 210501.

[15] Binder, Sven, et al. "Effective field theory in the harmonic oscillator basis." Physical Review C 93.4 (2016): 044332.

[16] Bansal, Aaina, et al. "Pion-less effective field theory for atomic nuclei and lattice nuclei." Physical Review C 98.5 (2018): 054301.

[17] Mbeng, Glen Bigan, Angelo Russomanno, and Giuseppe E. Santoro. "The quantum Ising chain for beginners." arXiv preprint arXiv:2009.09208 (2020).

[18] Furnstahl, R. J., S. N. More, and T. Papenbrock. "Systematic expansion for infrared oscillator basis extrapolations." Physical Review C 89.4 (2014): 044301.

[19] More, S. N., et al. "Universal properties of infrared oscillator basis extrapolations." Physical Review C 87.4 (2013): 044326.

[20] Coon, Sidney A., and Michael KG Kruse."Properties of infrared extrapolations in a harmonic oscillator basis." International Journal of Modern Physics E 25.05 (2016): 1641011.

[21] Gupta, Rajan. “Introduction to lattice QCD.” arXiv preprint hep-lat/9807028 (1998).

[22] Bansal, Aaina. “Medium Mass Nuclei Using Pionless Effective Field Theory in Harmonic Oscillator Basis.” (2019).