Simulation of Single-Particle Schrodinger Equation on Quantum Processor

BS PHYSICS

SUBMITTED BY

HAFSA ARSHAD

ID

F2017068011

DEPARTMENT OF PHYSICS
SCHOOL OF SCIENCE
UNIVERSITY OF MANAGEMENT AND TECHNOLOGY, LAHORE, PAKISTAN.

MARCH 2021

Simulation of Single-Particle Schrodinger Equation On Quantum Processor

A project submitted to the University of Management and Technology in accordance with the partial requirements for the award of the BS Physics degree

SUBMITTED BY

HAFSA ARSHAD

ID

F2017068011

DEPARTMENT OF PHYSICS
SCHOOL OF SCIENCE
UNIVERSITY OF MANAGEMENT AND TECHNOLOGY, LAHORE, PAKISTAN.

MARCH 2021

RESEARCH COMPLETION CERTIFICATE

This is clarified that the work material contained in project **Simulation of Single-Particle Schrodinger Equation on Quantum Processor** submitted by **Hafsa Arshad (F2017068011)** has been accepted. The quantum and quality of the work contained in this thesis is adequate for the award of degree of BS Physics.

.....	.....
Supervisor	**Examiner 3**
Hafiz Arslan Hashim	Ms Sidra Khalid
Lecturer	Assistant Professor
Department of Physics	Department of Physics
University of Management and Technology	University of Management and Technology
(UMT), Lahore, Pakistan	(UMT), Lahore, Pakistan
Examiners:	.....
	Dr. Badar Suleman
	Chairperson
.....	Department of Physics
Examiner 1	University of Management and Technology
Dr. Badar Suleman	(UMT), Lahore, Pakistan
Adjunct Professor	
.....	.....
Examiner 2	Dr. Muhammad Azhar Iqbal
Dr. Qayyum Zafar	**Dean**
Assistant Professor	School of Science (SSC)
	University of Management and Technology
	(UMT), Lahore, Pakistan

Abstract

Quantum computing uses superposition and entanglement to perform computation. During this project, we explored the basic quantum circuits, from a single qubit gate to Shor's factoring algorithm, and implemented them on a quantum processor by using Qiskit. As a main avenue, we studied the Schrodinger equation of a single free particle ($V(x) = 0$) in one dimension on a quantum processor. In particular, we implemented a 2-qubit quantum simulation algorithm on Qiskit to solve the Schrodinger equation. We showed the time evolution of the state and explained the particle's position in terms of probabilities for different qubit states.

Dedication and acknowledgments

I am dedicating this work to my late mother and living father who motivated me to give my best and after that to my teachers especially to my supervisor who inspired me to do hard work.

AUTHOR'S DECLARATION

I **Hafsa Arshad**, ID: **F2017068011**, student of **BS Physics**, secion **2017-2021** aware of and understand the university's policy on plagiarism. I certify that this thesis titled "**Simulation of Single-Particle Schrodinger Equation on Quantum Processor**" is my own work, expect where indicated by referencing, and the work presented in it has not been submitted in support of another degree or qualification from this or any other university or institute of learning. Due references have been provided on all supporting literatures and resources.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE **Page**

CHAPTER 1

INTRODUCTION

Quantum computation is based on the discovery that quantum systems are exponentially powerful. What does it mean? Suppose we have a system of 500 particles (for definiteness), which means its computing power can be as large as 2^{500}. We try to compute quickly by increasing the speed of computation of a computer or by having many computers working in parallel. If this computing power could be exploited in quantum mechanics, it will give us unimaginable fast computers. 2^{500} is impossible number. 2^{500} >> number of particles in the universe, 2^{500} >> age of universe in femtosecond, also greater then the product of both. The major goal of quantum computation is to harness this exponential power to solve computational problems (the most famous example is factoring). If it is permissible then there is no way in the classical universe that we could meet it even if the entire source of the universe in that computation could be approached [2]. Now of course the difficulty lies in harnessing this power. There are several challenges

- First, we have to pick the right computational problem. Not every computational problem can speed up by quantum computation.
- Even if we have the right quantum problem, designing quantum algorithm is a very tricky task. Quantum algorithms are very different from the classical algorithms.
- Experimental realization.

Quantum computers have also some limitations. We believe that still there are some problems that cannot be solved quickly even if quantum computers are being used [2]. For-instance, real-time problems (traffic control systems) cannot be solved by using quantum computers. It is of course a challenge for building quantum computers. To understand that what tye of problems we can solve on quantum computer, we should know the basic principles of quantum mechanics.

Quantum computer is constructed from quantum circuit comprising wires and elementary quantum gates to carry around and handle the quantum information [4].
Quantum state is the mathematical manipulation that grants us the probabilities of gaining a quantum system then we do measurement. Changes occurring to quantum states can be described using the language of quantum computing. The final aim of quantum computing is to build a machine based on quantum logic.
The task of this project is the simulation of a single particle Schrodinger equation on a quantum processor. In this example, the main characteristics of a quantum computer are discussed. We will regenerate this problem for 2-qubits from paper [1] to discuss the quantum logic where simulation of basic quantum mechanical problems is allowed. By using the Qiskit (python based library) on classical computer, we will access the actual quantum computer (ibmq×4) [1]. We will run the Schrodinger equation's simulation algorithm on ibmq×4 through Qiskit. Basically we will use Qiskit to built quantum circuit which will follow the quantum simulation steps. The understanding of python code and algorithm's implementation is developed from the paper at {`https://github.com/lanl/quantum_algorithms`}. Also some description of quantum circuits will help to built key institution of quantum computing.
Various aspects of quantum computing are also discussed in this project. Quantum Fourier Transform (QFT) is also explained and showed how it helps to build Shor's algorithm and many others. Also in this project, the source of this exponential power will be discussed and how to use this to design quantum algorithms. The strategy is to first pick the right computational problem then design a quantum algorithm.

CHAPTER 2

QUANTUM LOGICS

2.1 Qubits

Designing basic principles of quantum mechanics in terms of qubits greatly simplifies the demonstration. Another interest in studying quantum mechanics in this way is that we will be able to jump right into the most counter-intuitive aspect of quantum mechanics. Qubit is just the simplest description of the information in the classical world. The most straightforward definition of the qubit is defined in [3] as **"Physically, a qubit is a two-level system, like the two spin states of a spin-$\frac{1}{2}$ particle, the polarization states of a single photon or two states of an atom".** The Two features (1) coherent superposition (2) quantum entanglement are observed by quantum mechanic.

Using an electron to represent a bit: We can encode a bit by saying that ground state $\leftrightarrow$ $|0\text{range } arg1$ and excited state $\leftrightarrow |1\rangle$. The electron in an atom doesn't make up its mind whether it is in ground state or in excited state. Electron can be partly in the ground state and partly in the excited state. Electron is in some sort of superposition of ground and excited state, where each with some complex amplitude

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{2.1}$$

Another way we can represent qubit is

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2.2}$$

a unit vector in a 2-dimensional complex vector space. In this confused state which describes the state of an electron, these two complex numbers (α and β) carries an infinite number of bits of information. Complex amplitudes are normalized. However, when one looks at an electron,

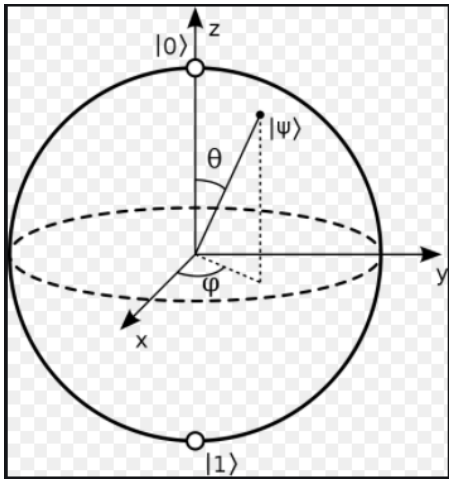

Figure 2.1: The state of the system (consists of complex amplitudes) on the Bloch sphere design in 3-dimensional real space. This figure is revived from {https://en.wikipedia.org/wiki/Bloch_sphere}.

it quickly makes up its mind called measurement. And it goes into either ground state or excited state with some probabilities. This is the reason for the state to be normalized because probabilities must add up to 1. Pure state represented by 2-D complex vector with a length of one which is normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.3}$$

A real-world example of a qubit: Photons have a property called polarization which caries a qubit of information. Felix Bloch had a very good idea on how to visualize the state of a qubit as a point in 3-D space, more precisely as a point on the sphere which is nowadays called as Bloch sphere. Bloch's contribution to quantum computing was that he wrote complex amplitudes in exponential form. We can use spherical coordinates to write qubit $|\psi\rangle$ in only two real numbers θ and φ represented as $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{\iota\varphi}\sin\frac{\theta}{2}|1\rangle$. Here $e^{\iota\varphi}$ tell us about relative phase, this is how we get the interference effect. Every spot on the Bloch sphere corresponds to exactly one possible state of a qubit [4]. And range for $\theta \in [0, 2\pi)$, here 2π is excluded from range since it compares to 0 and $\varphi \in [0, \pi]$.

For one qubit: In Fig. 2.1 the "north pole" represents state $|0\rangle$, the south pole represents state $|1\rangle$ on sphere. The size of the blob is equivalent to the probability of measuring the respective state. Bloch sphere can only illustrate the state of a single qubit. We use Q-sphere to visualize the state of multiple qubits. For n-qubits, we have 2^n basis states, and these basis states are plotted as equally distributed points on a sphere with $0^{\otimes n}$ on the z-axis, $1^{\otimes n}$ on the negative z-axis, and all other states are regulated on parallel such that the number of ones on each latitude is constant and increasing from north to south pole. Here in Fig. 2.1, θ is the latitude which determines the probability that state either $|0\rangle$ or $|1\rangle$ and φ is longitude. For instance if $\theta = 0°$, the quantum bit is in basis state $|0\rangle$. If $\theta = 180°$, the quantum bit is in basis state $|1\rangle$.

Be-careful: On the Bloch sphere, angles are twice as in Hilbert space e.g $|0\rangle$ and $|1\rangle$ are orthonormal but on the Bloch sphere, their angle is π. θ is the angle on the Bloch sphere while $\theta/2$ is the actual angle in Hilbert space. The state $|\psi\rangle$ on Bloch sphere in Fig. 2.1 is state before measurement.

2.2 Axioms of Quantum mechanics

2.2.1 Superposition

The unit vector is in a k-dimensional complex vector space (Hilbert space). It tells us about k-level system (for example k-energy levels) of an electron, the state of the system is a point on the k-D ball in k-D complex space. Meaning that there is a k-D complex vector space with selected basis which consists of these k-states labeled from $|0\rangle$ to $|k-1\rangle$ as

$$\left|\psi\right\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + ... + \alpha_{k-1} |k-1\rangle . \tag{2.4}$$

Superposition allows us performing the calculation on many states at the same time as apparent in the above equation. $\left|\psi\right\rangle$ is a unit vector in space, design in matrix form is

$$\left|\psi\right\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ . \\ . \\ . \\ \alpha_{k-1} \end{bmatrix} . \tag{2.5}$$

Complex vector space has the notation of orthogonality between these states and there is an angle between other states which is defined by an inner product. So this complex vector space with an inner product is called Hilbert space. Anything we can compute classically ia also possible in superposition [4].

2.2.2 Measurement

A measurement is specified by choosing an orthonormal basis $|U_0\rangle, |U_1\rangle, ..., |U_{k-1}\rangle$ which form the state

$$\left|U_j\right\rangle = \beta_0 |U_0\rangle + \beta_1 |U_1\rangle + ... + \beta_{k-1} |U_{k-1}\rangle . \tag{2.6}$$

The probability of each outcome is the square of the length of the projection onto the corresponding observed basis vector

$$\left|\psi\right\rangle = \alpha_0 \left|\psi_0\right\rangle + \alpha_1 \left|\psi_1\right\rangle + ... + \alpha_{k-1} \left|\psi_{k-1}\right\rangle . \tag{2.7}$$

Once we measure the superposition state, it collapses to the observed basis vector. So, we get only one state

$$\text{inner product} = \langle U_j|\psi\rangle, \tag{2.8}$$

$$p_r[j] = |\langle U_j|\psi\rangle|^2. \tag{2.9}$$

New state is $|\psi'\rangle = |U_j\rangle$.

2.2.3 Quantum entanglement

Entanglement discovered by Einstein is the key resource that makes exponential speed-ups possible for quantum computation and quantum information. Classically, we can represent 2-qubits of information by superposition the quantum state of these two electrons, in general of all four possibilities. The state for 2-qubits is $\left|\phi\right\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$ and this state should fulfill condition $\sum|\alpha|^2 = 1$. The composite state says that electrons don't make up their mind that they are in which of the four classical states.

When we do measurement, it disturbs the system meaning that electrons quickly make up their mind whether they are in ground or excited state. For instance, if we measure qubit state, the probability of measuring the $|0\rangle$ is the same as for $|1\rangle$. There is no way to factor composite state as a state of the first qubit times the state of the second qubit. Another way to say is, quantum systems have the property that if two quantum systems come close together and interact with each other then they get into a state. Where neither of these two quantum systems can be specified by itself. They get so entangled with each other, the only way to explain the system is by calling it collectively and can't explain it by calling each of these particles separately. Now it turns out that this entangled state of two quantum systems or quantum particles persists. Even if letting them interact with each other, we now separate them by a great distance. So they stay entangled even though they are very far from each other. If something interacts with just one part of our system then our whole system is affected.

Einstein explained that particles contained hidden information which we didn't know before measurement. That information was within the particle from the moment they formed at some point in space. No signal would ever have to travel between two particles faster than light. Entangled particles can signal each other faster than light to update their hidden information when one is measured but it doesn't mean that we can use entangled particles to communicate faster than the speed of light.

Pure quantum states correspond to vectors in Hilbert space. Correlated states are not entangled states but they are mixed states. The pure state is correlated and also entangled. If a pure state $|\psi\rangle_{AB}$ of system A,B can't be written as $|\psi\rangle_A$ and $|\phi\rangle_B$, it is entangled. There are four so-called Bell states that are maximally entangled and built on an orthonormal basis:

$$|\psi^{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{2.10}$$

$$|\psi^{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{2.11}$$

$$|\psi^{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{2.12}$$

$$|\psi^{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{2.13}$$

In general set of four bell states can be written as

$$|\psi^{ij}\rangle = (I \otimes \sigma_x^i \sigma^i{}_z)|\psi^{00}\rangle. \tag{2.14}$$

If the state is $|\psi^{01}\rangle$, by measuring 1st qubit we get $|0\rangle$, then immediately we get to know the 2nd qubit will be $|1\rangle$. I have reproduced concepts about quantum entanglement from Umesh Vazirani's lectures available at {`https://people.eecs.berkeley.edu/~vazirani/quantum.html`}.

2.2.3.1 EPR Paradox

EPR paradox was invented by Einstein, Podolsky, and Rosen in 1935 to express dissatisfaction with quantum mechanics. Bell state in sign basis $|\psi\rangle = \frac{1}{\sqrt{2}}|--\rangle + \frac{1}{\sqrt{2}}|++\rangle$, entangled state of two qubits. By expanding in superposition of $|00\rangle$ and $|11\rangle$, it comes out to be $\frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$. This means that when two qubits interact with each other, maybe with probability $\frac{1}{2}$ they both went into the state $|00\rangle$ and with probability $\frac{1}{2}$ they both went into $|11\rangle$. EPR paradox said, suppose these two qubits are very far from each other so that 1st qubit can't affect the 2nd qubit at least for a very long period. We can choose to measure qubit on a bit basis or sign basis. Whichever one is decided to be measured, the value of 2nd qubit is known. Here is an unusual thing is that the 2nd qubit is not affected by which one you decided to measure. EPR argues: what it means that both values of the 2nd qubit must be determined. This seems to contradict the uncertainty principle. It says, even though it is true if the bit value of 1st qubit is measured, it disturbs the sign value. But the 2nd qubit is far apart that measuring the bit value of 1st qubit doesn't change the sign value of 2nd qubit.
EPR concluded: Quantum mechanics is an incomplete theory when it tells that you can't measure both these values simultaneously meaning that in quantum mechanics we can't get values simultaneously. Quantum mechanics limits the amount of information that we can get about nature. State of 2nd qubit is not defined because itself is entangled with 1st qubit. Bit and sign values of 2nd qubit are well defined, it just doesn't make sense. This explanation in terms of ERR paradox is better at {`https://people.eecs.berkeley.edu/~vazirani/quantum.html`}.

2.2.4 Evolution

The Changes occurring in quantum state can be described using the language of quantum computing. Evolution tells us how the state of the system evolves. The evolution of the system is via a rotation of Hilbert space. By rotating complex vector space, everything inside this vector space rotates as a result. This is a rigid body rotation. Angles between vectors are preserved.
Rotation of complex vector space is linear transformation represented by a matrix, known

as unitary transformation (linear property). For instance, under this unitary transformation, state $|0\rangle$ maps to new state $(\cos\theta|0\rangle + \sin\theta|1\rangle)$, similarly $|1\rangle \rightarrow (-\sin\theta|0\rangle + \cos\theta|1\rangle)$. Unitary transformation in matrix form

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{2.15}$$

Rotation through $-\theta$ is exactly the rotation transpose. Here the matrix

$$R_{-\theta} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{2.16}$$

is for 2 qubits. By rotating state with θ then again rotate by $-\theta$ one will come back to same initial state as:

$$R_\theta R_{-\theta} = R_{-\theta} R_\theta = I, \tag{2.17}$$

where R_θ & $R_{-\theta}$ are orthogonal to each other. This means that length is given by inner product with itself. Claim: $R_\theta\langle 0|\psi\rangle = R_\theta\langle\psi|0\rangle$, this shows that R_θ preserves inner product meaning that it always preserves angles between vectors.

CHAPTER 3

QUANTUM CIRCUITS AND QUANTUM ALGORITHMS

3.1 Quantum circuits

Quantum logic gates are reversible due to the coherence nature of quantum mechanics. While classical logic gates are not reversible since we can't ascertain input from the output. Quantum circuits can perform all functions that can be calculated classically. We know the quantum theory is the unitary theory, so unitary gates are considered only. They are reversible and follow the condition $U^\dagger U = I = UU^\dagger$ also $U^{-1} = U^\dagger$ [1]. Unitary gates denoted as U_θ are single qubit gates and these gates have a single input and single output. When one applies a unitary operator which means phase has effectively applied to the eigenvector. Predominantly, the results we get from unitary gates are the coherent superposition of $|0\rangle$ and $|1\rangle$

$$U_\theta|0\rangle = cos\theta|0\rangle + sin\theta|1\rangle, \tag{3.1}$$

$$U_\theta|1\rangle = sin\theta|0\rangle - cos\theta|1\rangle. \tag{3.2}$$

Particular case: Hadamard gate is a single qubit gate, result of maximum coherence between $|0\rangle$ and $|1\rangle$. Here, $|0\rangle$ maps to $|+\rangle$ and $|1\rangle$ maps to $|-\rangle$. Whereas $|+\rangle$ and $|-\rangle$ are sign basis and denoted as equal superposition of $|0\rangle$ and $|1\rangle$ with opposite phases

$$H|0\rangle = U_{\frac{\pi}{4}}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{3.3}$$

$$H|1\rangle = U_{\frac{\pi}{4}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{3.4}$$

Hadamard gate is used to create entanglement in some cases. This gate can be used in more than one qubit case $|00\rangle \rightarrow |++\rangle$ & $|11\rangle \rightarrow |\ldots\rangle$ and same for n-qubits. For $\theta = \frac{\pi}{2}$ **bit flip gate**,

denoted by X, is acquired ($|0\rangle \rightarrow |1\rangle$) and ($|1\rangle \rightarrow |0\rangle$) i.e.

$$X|0\rangle = U_{\frac{\pi}{2}}|0\rangle = |1\rangle, \tag{3.5}$$

$$X|1\rangle = U_{\frac{\pi}{2}}|1\rangle = |0\rangle. \tag{3.6}$$

Another single qubit gate is **Z-gate** which changes sign only when input is $|1\rangle$ and transformation at $\theta = 0$ in Eq. (3.1) & (3.18) is as

$$Z|0\rangle = U_0|0\rangle = |0\rangle, \tag{3.7}$$

$$Z|1\rangle = U_0|1\rangle = -|1\rangle. \tag{3.8}$$

Entangled states are built between two qubits using two-qubit gates. Two qubit gates are also reversible and have 2 inputs and 2 outputs.

Quantum phase gate Q_ϕ: Output transformation concerning two inputs is $Q_\phi|0_1,0_2\rangle = |0_1,0_2\rangle$, $Q_\phi|0_1,1_2\rangle = |0_1,1_2\rangle$, $Q_\phi|1_1,0_2\rangle = |1_1,0_2\rangle$, $Q_\phi|1_1,1_2\rangle = e^{\iota\phi}|1_1,1_2\rangle$. This gate does not change output if any of the two-qubit is $|0\rangle$, phase change occurs only when both qubits are $|1\rangle$ at the same time for input.

Controlled Not gate: It's a quantum NOT gate which consists of controlled bit and target bit for 1st and 2nd qubit respectively. If the controlled bit is set to $|1\rangle$ then it flips the target qubit otherwise it does nothing. CNOT gate process as $U_{CNOT}|0_1,0_2\rangle = |0_1,0_2\rangle$, $U_{CNOT}|0_1,1_2\rangle = |0_1,1_2\rangle$, $U_{CNOT}|1_1,0_2\rangle = |1_1,1_2\rangle$, $U_{CNOT}|1_1,1_2\rangle = |1_1,0_2\rangle$. CNOT gate just does not appear in a 2-qubit computation, it might appear in many qubit computations.

Quantum circuits build from quantum gates. Quantum computers are built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate quantum information. Quantum algorithms consist of three basic steps [1]:

- Encoding data, which could be classical or quantum, into the state of a set of input qubits.
- A sequence of quantum gates applied to this set of input qubits.
- Measurement of one or more qubits at the end to obtain a classically interpretable result. A quantum algorithm is just like a set of instructions that we give to a quantum computer, means it tells the computer how to behave.

3.2 Quantum Fourier Transform

When elementary quantum gates are used to perform the unitary transformation, it is not systematic since it requires the $\bigcirc(n^2 4^n)$ number of quantum gates. A more structured way to perform this kind of transformation is through the quantum circuit model namely the number of quantum gates are n-polynomial. To build the Quantum Fourier transform (QFT) circuit, the following 2 steps needed to go through: (1) Show QFT rigorously (2) Show circuit to implement

QFT expression that comes from (1). The QFT is used to transform n-qubit states linearly from computational to Fourier basis states

$$|x'\rangle \equiv QFT|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=o}^{N-1} e^{\frac{2\pi\iota xy}{N}} |y\rangle, \tag{3.9}$$

$|x'\rangle$ represents Fourier basis and computational basis are $|y\rangle = \sum_{k=1}^{n} y_k 2^{n-k} = |y_1, y_2, \ldots, y_n\rangle$. The QFT is exactly the inverse of the discrete Fourier transform. n-qubits have $N = 2^n$ basis states on which QFT is performed. Hadamard gate performs the same QFT for single-qubit in which Z-basis transforms into X-basis ($|+\rangle$ & $|-\rangle$)

$$QFT|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\iota\pi 0}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \tag{3.10}$$

$$QFT|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\iota\pi 1}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle, \tag{3.11}$$

where $|+\rangle$ & $|-\rangle$ are Fourier basis and computational basis states are $|0\rangle$ and $|1\rangle$. Now above equation becomes

$$= \frac{1}{\sqrt{N}} \sum_{y=o}^{N-1} e^{2\pi\iota x \sum_{k=1}^{n} \frac{y_k}{2^k}} |y_1, y_2, \ldots, y_n\rangle, \tag{3.12}$$

$$= \frac{1}{\sqrt{N}} \sum_{y=o}^{N-1} \Pi_{k=1}^{n} e^{2\pi\iota x \frac{y_k}{2^k}} |y_1, y_2, \ldots, y_n\rangle. \tag{3.13}$$

After the application of QFT to the computational basis:

$$|x\rangle = |x_1 x_2 x_3 \ldots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes |x_3\rangle \otimes \cdots \otimes |x_n\rangle, \tag{3.14}$$

we get the Fourier basis:

$$|x'\rangle = \frac{1}{\sqrt{N}}(|0\rangle + e^{\frac{2\pi\iota x}{2^1}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi\iota x}{2^2}}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{\frac{2\pi\iota x}{2^n}}|1\rangle). \tag{3.15}$$

In the above expression, 1st term represents the transformation of 1st qubit, 2nd term shows the transformation of 2nd qubit, and so on. Notice that each qubit has a different phase. Two observations from the above expression help us to build the QFT circuit:

- $|x'\rangle$ contains term like $|000\ldots0\rangle$, $e^{\frac{2\pi\iota x}{2^n}}|000\ldots1\rangle$, $e^{\frac{2\pi\iota x}{2^{n-1}}}|000\ldots10\rangle$ and so on. The power in exponential terms $2^n, 2^{n-1}, \ldots, 2^0$ and the pattern in which qubits flipped to $|0\rangle$ to $|0\rangle$ in $|00\ldots00\rangle$, $|00\ldots01\rangle$, $|00\ldots10\rangle$, …, $|10\ldots0\rangle$ are tied together. When a qubit is flipped from $|0\rangle$ to $|1\rangle$, the phase seems to change. This shows that the phase is qubit-dependent. This help us to construct a quantum circuit for QFT.

- Combined form of Eq. (3.15) is

$$|x'\rangle = e^{\frac{2\pi\iota x}{2^1} + \frac{2\pi\iota x}{2^2} + \cdots + \frac{2\pi\iota x}{2^n}} |1111\ldots1\rangle). \tag{3.16}$$

  The phase $e^{2\pi\iota[\frac{x}{2^1} + \frac{x}{2^2} + \cdots + \frac{x}{2^n}]}$ is in exponential power in front of state of all $|1\rangle$'s. Notice that when the state has $|0\rangle$'s it has no phase, by adding more $|1\rangle$'s in phase we have to add more components in phase. This is the key intuition that allows constructing a quantum circuit.

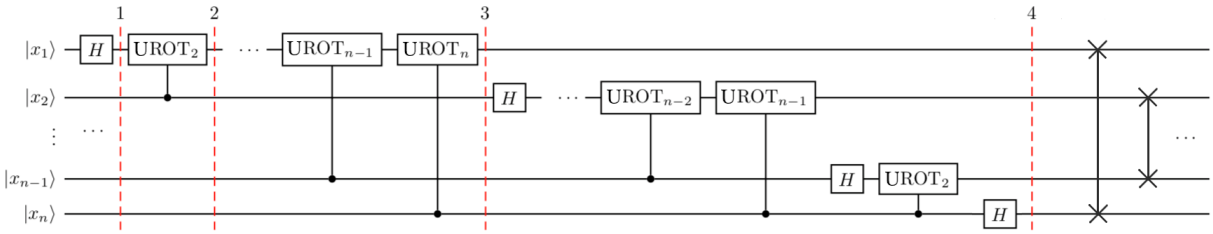

Figure 3.1: Quantum circuit of QFT. I have regenerate this circuit from {`https://qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html`}.

Now we can use Eq. (3.15) to built a quantum circuit. It needs two ingredients:

- 1st thing one can think of is Hadamard gate which is applied on particular qubit with that generic x_k and do something like $H|x_k\rangle = [|0\rangle + e^{\frac{2\pi i x_k}{2}}|1\rangle]/\sqrt{2}$. $e^{\frac{2\pi i x_k}{2}}$ gives phase that looked like 2^1 (in denomerator) means only on one qubit.

- 2nd gate will use called U rotation that gives a little bit of information about the value of k. when we apply it to a particular qubit state $|x_j\rangle$ as

$$UROT_k|x_j\rangle = e^{\frac{2\pi i x_j}{2^k}}|x_j\rangle, \tag{3.17}$$

  this rotation depends on the number of qubits. Nothing will change if the qubit is $|0\rangle$ or $|1\rangle$. For single qubit, $UROT_k$ is 2×2 matrix

$$UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i x_k}{2^n}} \end{bmatrix}, \tag{3.18}$$

  and job of this matrix is to apply phase $e^{\frac{2\pi i x_k}{2^n}}$ for state $|1\rangle$.

In Fig. 3.1, Qubit-1 seems like the control coming from 2nd qubit, 3rd qubit, and all other qubits down to n-qubits. For the 2nd qubit, control is coming from everything below it, it never sees that control comes from qubit-1. The state is $|x_1x_2\ldots x_n\rangle$. UROT is being apply on $|x_1\rangle$ as shown: $[|o\rangle + e^{\frac{2\pi i x_1}{2^1}}|1\rangle] \otimes |x_2x_3\ldots x_n\rangle$, remember UROT action on 1st qubit. It does the same for state $|x_2\rangle$ and up to n-qubits as $[|o\rangle + e^{\frac{2\pi i x_1}{2}} e^{\frac{2\pi i x_2}{2^2}} e^{\frac{2\pi i x_3}{2^3}} \ldots e^{\frac{2\pi i x_n}{2^n}}|1\rangle] \otimes |x_2x_3\ldots x_n\rangle$. In Eq. (3.15) step-n term was the last term which is for $|x_1\rangle$ but in our circuit it is the 1st term $|x_1\rangle$. There is a reverse order in which the circuit works. But when we implement it, we see that it's the same as what we did with QFT. These steps are only for 1-qubit, we can do the same steps for n-qubits. This circuit implements QFT except in reverse order of qubit of output. This means we started with the general representation of QFT and this is exactly the reverse of what this circuit does. QFT is a way of building our Shor's algorithm and finding periodicity in some way [3].

3.3 Quantum Phase Estimation (QPE) Algorithm

Now we show the QPE algorithm implementation and that how QFT fits into this big picture.
Problem: Remember that the unitary matrix has eigenvalues of the form $e^{\iota\theta}$ and it has eigenvectors that form an orthonormal basis.
Objective: If unitary U is given and applied on the state $|\psi\rangle$

$$U|\psi\rangle = e^{\iota\theta_\psi}|\psi\rangle, \tag{3.19}$$

then the operation of U on a state $|\psi\rangle$ is to apply some phase that depends on eigenstate or eigen vector $|\psi\rangle$. The QPE gives $2^n\theta$, where n is the number of qubits used to estimate θ. If we have some accessible that is able to prepare $|\psi\rangle$ and to apply U itself then we can extract θ_ψ. Here the goal is to find the global phase θ_ψ. QPE allows us to convert this phase information into things that can be measured.
For instance, the state is $e^{\frac{\iota\pi}{2}}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and results after measurement are $|e^{\frac{\iota\pi}{2}}/\sqrt{2}|^2 = \frac{1}{2}$ for $|0\rangle$ and same for $|1\rangle$. This shows the difficulty of applying the global phase here and measuring it out. If the global phase $e^{\iota\theta}$ was applied by the unitary operator. Unitary and Hermitian matrices are related to each other. The kind of things, that we can measure, that represent physical systems are Hermitian operators and the time evolution of those systems is unitary. So, Hermitian evolution (time evolution of real system) is unitary. This has implication for quantum simulation. Suppose if $\theta_\psi = 1$ (phase applied by unitary) then the probabilities of measuring $|0\rangle$ and $|1\rangle$ are 0.9999 and 7.6×10^{-5} respectively. In the difference between the probability of measuring $|0\rangle$ and $|1\rangle$, phase is encoded. In other words, the phase information has taken and turned into an amplitude that can be measured. Painful experiment: Millions of shots needed to do experiment then statics would be collected and see how many times we get $|0\rangle$ and $|1\rangle$. The difference in statistics between $|0\rangle$ and $|1\rangle$ would allow back to θ.
$Bettersetup$: To do much more precise measurement, use QFT by using more qubits. We have used 1-qubit and the information (the amplitude) is coming from that qubit for us to understand what the phase is. An expression for QPE:

$$|\psi''\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \left(|0\rangle + e^{\iota\theta_\psi 2^{n-1}}|1\rangle\right) \otimes \left(|0\rangle + e^{\iota\theta_\psi 2^{n-2}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{\iota\theta_\psi 2^0}|1\rangle\right)|\psi\rangle. \tag{3.20}$$

By comparing this expression with Eq. (3.15), it is find out that QPE is same as QFT except $\theta_\psi \to 2\pi\left[\frac{\theta_\psi}{2^n}\right]$. All of this is exactly writing out QFT with different phase $2\pi\left[\frac{\theta_\psi}{2^n}\right]$. The expected state is $|2^n\theta_\psi\rangle$ is because of mismatch between $\theta_\psi \to 2\pi\left[\frac{\theta_\psi}{2^n}\right]$. This is the QFT of something and then we apply QFT of exact expression which end up giving $|2^n\theta_\psi\rangle$. The reason to came down this path is that there was one qubit on the set of qubits and we didn't have enough precision to do our calculation. As θ is very small, in this particular case, phase is multiplied by 2^n, where n is the no. of qubits to estimate θ. To get rid of lots of shots and multiplying that phase we are looking for 2^n which gives precision.

QPE protocol: tells if the sequence of gates has applied at the fact QFT containing information about phase and its amplitude. Then apply QFT inverse ($QFT^{\dagger}$), which means trying to convert phase information back into amplitude. In circuits $QFT^{\dagger}$starts working from right side to left side in reverse, keeping in mind that we are using complex conjugate of these operations not the same operation.
QFT is also a way of building Shor's algorithm and finding periodicity in some way. QFT can serve in different ways but the most popular one is to solve the following **problem:** If a periodic function is given then we find its period. The goal of **Shor's algorithm:** Factoring a number $N = pq$ where p and q are prime and large numbers. This algorithm has reproduced from the book of Qiskit community whose description is given at {https://www.youtube.com/watch?v=5kcoaanYyZw&list=PLOFEBzvs-VvrXTMy5Y2IqmSaUjfnhvBHR&index=10}.

CHAPTER 4

QUANTUM SIMULATION

Simulation of physical systems is the accomplishable example of quantum computation. The main task of simulation is to find the solution of physical laws-based differential equations, for instance, Newton's law, Poisson's equation, diffusion equation, etc. [4]. The simulation starts with some initial state and then has to find the state with the evolution of time and space. Approaching the desired state with digital characterization, and doing the number of iterations on the differential equations (mean discretization) gives final results. On classical computers, quantum simulation is possible but it is not that efficient. Solving the number of exponential differential equations is the main difficult task in quantum simulation. For example, n-qubits require 2^n number of equations to solve. Less number of equations can be simulated to make the quantum system achievable by introducing some approximations. Classical simulation is not helpful for most quantum systems. Simulation for quantum systems can be done on quantum computers. Same as universal quantum gates can be used to built quantum circuits since they perform unitary operations but they needed no approximations. Such systems are Hamiltonian base quantum systems.

Quantum simulation of Schrodinger equation: Consider a single particle, having some potential which is commanded from Hamiltonian [3]. We have demonstrated such type of system by using quantum simulation. Quantum simulation is started by solving the differential equation. Schrodinger equation in its time-dependent version

$$\imath\hbar\frac{d}{dt}|\psi(x,t)\rangle = H|\psi(x,t)\rangle. \tag{4.1}$$

Hamiltonian gives us the total energy built mainly out of two parts (K.E and P.E)

$$H = K.E + V(x) = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x) = \frac{p^2}{2m} + V(x). \tag{4.2}$$

In general, K.E part is written with momentum as shown in Eq. (4.2), which tells us that how fast the system moves and potential is the function of our x-coordinate in classical Cartesian space. If Eq. (4.1) gets integrated for time-dependent Hamiltonian, we get an equation which tells us that if we apply $e^{\frac{-\iota Ht}{\hbar}}$ to state at time zero we get the state at time t

$$|\psi(t)\rangle = e^{\frac{-\iota Ht}{\hbar}}|\psi(0)\rangle. \tag{4.3}$$

This behavior is as it propagates wave function from time $0 \rightarrow t$, that's why $e^{\frac{-\iota Ht}{\hbar}}$ known as a propagator. In Eq. (4.1) position x and time t both are continuous which needed to be discretized. This requirement is because the solution is obtained by using the quantum computer which consists of a finite number of gates and qubits. 2^n intervals with specific length $\triangle = \frac{2d}{2^n}$ needed if the single particle is moving in region $-d \leq x \leq d$. The state occupies in Hilbert space with basis x for n-qubits

$$|\psi\rangle = \int_{-\infty}^{\infty} |x\rangle\langle x|\psi\rangle dx. \tag{4.4}$$

$\triangle x$ is the discretization step that must be small in comparison with the system's wavelength. The approximate outcome[4] for state $|\psi\rangle$

$$|\psi\tilde{}\rangle = \sum_{k=-d/\triangle x}^{d/\triangle x} a_k |k\triangle x\rangle. \tag{4.5}$$

The number of qubits required to demonstrate the state are $n = [log(2d/\triangle x + 1)]$. Now for n-qubits, $|k\rangle = |k_{n-1}\rangle|k_{n-2}\rangle...|k_0\rangle$ is computational basis state that comes in replacement with the basis state $|k\triangle x\rangle$. To do simulation on quantum computer only n number of qubits are required and the complex numbers defining state classically are 2^n.
In Eq. (4.3) its difficult to explain the term $e^{\frac{-\iota}{\hbar}[H]t}$. For quantum systems, we can use approximation to compute $e^{\frac{-\iota}{\hbar}[H]t}$, but $[K.E, V] \neq 0$. We can separate the exponential terms, if the time is discretized in small steps

$$e^{\frac{-\iota}{\hbar}[K.E+V(x)]t} \approx e^{\frac{-\iota}{\hbar}K.E\triangle t} e^{\frac{-\iota}{\hbar}V(x)\triangle t}. \tag{4.6}$$

Above expression is known to be Trotter decomposition and still perform the unitary operation [4]. To get more precision in results, we don't do huge steps from 0 to t. If the system goes in that way from 0 to t in small steps, we can write small time steps $\epsilon = \frac{t}{N}$ as

$$e^{\frac{-\iota Ht}{\hbar}} = \left[e^{\frac{-\iota Ht}{N\hbar}}\right]^N = \left[e^{\frac{-\iota H\epsilon}{\hbar}}\right]^N. \tag{4.7}$$

After doing this N-time, it still doesn't help us since we don't know how to compute the application of this operator to some general state. The Hamiltonian can be written in two parts (K.E and P.E part). But we cannot write this propagator as a product of two, one for the K.E part and other for the P.E part. The world is not that bad, Schrodinger gave us an equation that help us to do time evolution in quantum mechanics

$$e^{-\iota\frac{(K.E+V)}{\hbar}t} = \lim_{N\rightarrow\infty}\left[e^{\frac{-\iota Vt}{2N\hbar}} e^{\frac{-\iota K.Et}{N\hbar}} e^{\frac{-\iota Vt}{2N\hbar}}\right]^N. \tag{4.8}$$

Here, the half time step is in potential, full-time step in kinetic, and again half time step in potential. And limit in Eq. (4.8) shows that we repeat this expression an infinite number of times. This is a controllable approximation which gets better if we use more time steps.
In Eq. (4.4), the term $\langle x|\psi\rangle$ shows that state $|\psi\rangle$ project into a coordinate basis as

$$\langle x_t|\psi(x_t)\rangle = \int \langle x_t|e^{\frac{-\iota Ht}{\hbar}}|x_0\rangle\langle x_0|\psi(0)\rangle dx_0. \tag{4.9}$$

Now we can examine from Eq. (4.3) that we start with wave function at time zero $|\psi(x,0)\rangle$. After doing some algebra, the equation Eq. (4.4) becomes

$$|\psi(x_1,\epsilon)\rangle = e^{-\frac{\iota V(x_1)\epsilon}{2\hbar}}\left(\int \frac{dp}{\sqrt{2\pi\hbar}}e^{\frac{\iota px_1}{\hbar}}\right)e^{-\frac{\iota p^2\epsilon}{2m\hbar}}\left(\int \frac{dx_0}{\sqrt{2\pi\hbar}}e^{-\frac{\iota px_0}{\hbar}}\right)e^{-\frac{\iota Vx_0\epsilon}{2\hbar}}|\psi(x_0,0)\rangle. \tag{4.10}$$

The above method is the split operator method which is taken from {`https://www.youtube.com/watch?v=BBt8EugN03Q`}. Using some approximations used in Eq. (4.6), Eq. (4.7), Eq. (4.8) regarding propagator, we get the state after N-time steps. The terms in the brackets refers to QFT. We start with the computational basis then we transform the state to momentum basis through QFT. In momentum basis, we apply kinetic propagation which has p (momentum) in it. After that, we transform this from momentum basis to computational basis with the help of $QFT^\dagger$. And lastly, potential propagation is again applied to complete the time step. The operator in the left side of Schrodinger Eq. (4.1) can be written in terms of QFT and $QFT^\dagger$ as

$$-\iota\hbar\frac{d}{dx} = QFT^{-1}pQFT, \tag{4.11}$$

or

$$e^{\frac{-\iota}{\hbar}K.E\triangle t} = QFT^{-1}e^{\frac{-\iota}{\hbar}(\frac{p^2}{2m})\triangle t}QFT. \tag{4.12}$$

Here the potential is harmonic oscillator $V = \frac{1}{2}m\omega^2x^2$ which is explained in Ref.[3] and we are rewriting it. Also, it can be seen by discretizing the variable x in $\triangle t$ time steps as

$$x = \alpha\sum_{j=0}^{n-1}(k_j2^j + \beta), \tag{4.13}$$

$$x^2 = \alpha^2\sum_{j,l=0}^{n-1}(k_j2^j + \beta)(k_l2^l + \beta), \tag{4.14}$$

and the potential propagator decomposed as

$$e^{-\frac{\iota V(x)\epsilon}{\hbar}} = \Pi_{j,l=0}^{n-1}e^{-\iota\gamma(k_j2^j+\beta)(k_l2^l+\beta)}, \tag{4.15}$$

where $\gamma = \frac{m\omega^2\alpha^2\epsilon}{2\hbar}$, $\beta = \frac{-d+\triangle/2}{\triangle n}$, and $\alpha = \triangle$. For momentum basis, Hamiltonian is proportional to p^2 and the decomposition of kinetic energy propagator is shown in equation Eq. (4.12).
If we have an initial state of wave function $|\psi(x,0)\rangle$ and by applying the unitary operator

$$e^{-\iota V(x_i)\epsilon}QFT^\dagger e^{-\frac{\iota}{\hbar}\left(\frac{p^2}{2m}\right)\epsilon}QFT, \tag{4.16}$$

finite number of time (l time), we can get the state $|\psi(x_i,\epsilon)\rangle$ at time $t = \epsilon l$. Now from above whole discussion, we can summarize the simulation of a Schrodinger equation as the implementation of sum of QFT and the operator

$$|x\rangle \rightarrow e^{\iota f(x)} |x\rangle. \tag{4.17}$$

Our main interest is to find out the results for $|\psi(x_i,t)|^2$.

4.0.1 Algorithm description

Algorithm for simulating the Schrodinger equation of a single particle is followed from [1].

Algorithm 1 Schrodinger equation's simulation

Input:

- Initial state $|\psi(x,0)\rangle$ in computational basis.
- $t_f = \epsilon l$ where ϵ is the descretized time step.
- In computational basis the competency to bid phase shifts.
- The potential function V.

Output:

- The state of wave function $|\psi(x,t_f)\rangle$ at time t and have potential V.

Procedure

- The state of normalized wave function encrypted on N-point grid for $n = log_2 N$ qubits. When the wave function gets discretize then it has to be re-normalized.
  For $1 \leq j \leq t$
  Step-a Apply the quantum Fourier transform into our momentum basis.
  Step-b Apply the K.E propagator to state as $|x\rangle \rightarrow e^{-\iota x^2 \epsilon} |x\rangle$.
  Step-c Go back to coordinate representation via an inverse Fourier transform.
  Step-d Apply the last half time step for our potential propagation as $|x\rangle \rightarrow e^{-\iota V(x)\epsilon}|x\rangle$.
- Do measurements on a computational basis.

Now in order to simulate the Schrodinger equation, we don't need long steps. In order to find the propagation to an arbitrary functions, a series of the CNOT gates is used. The CNOT gates create entangled states that's why named as entangled gates. Approximate amplitudes of wave function $|\psi(x,t_f)|^2$ can be found by repeating the last step listed in the procedure over number of times. I have used 2-qubits to solve the Schrodinger equation on quantum computer and I have considered the single free particle i.e $V(x) = 0$. Rectangular function (Π function) is initial wave function

$$|\psi(x,0)\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \tag{4.18}$$

Its a Bell state for $\phi = 0$. For 2-qubits this wave function depicted as 0,1,1,0 on 2^5 point-grid. Also in Eq. (4.16) I have taken value of $\hbar$ equal to 1 and $m = \frac{1}{2}$. Now the iterative equation becomes

$$|\psi(x_i, t_f)\rangle = QFT^{\dagger} e^{-\iota P^2 t_f} QFT |\psi(x_i, 0)\rangle. \tag{4.19}$$

QFT can be decomposed as a product of unitary gates [3]. For 2-qubit, $QFT = SWAP_{12} H_2 C(P_1(\frac{\pi}{2}))H_1$ where C_2P shows controlled phase rotation. The momentum represented as

$$P = -\frac{1}{2}\sqrt{\frac{\phi}{\epsilon}}\,(1 + \sum_{P=1}^{n} 2^{n-P} Z_P), \tag{4.20}$$

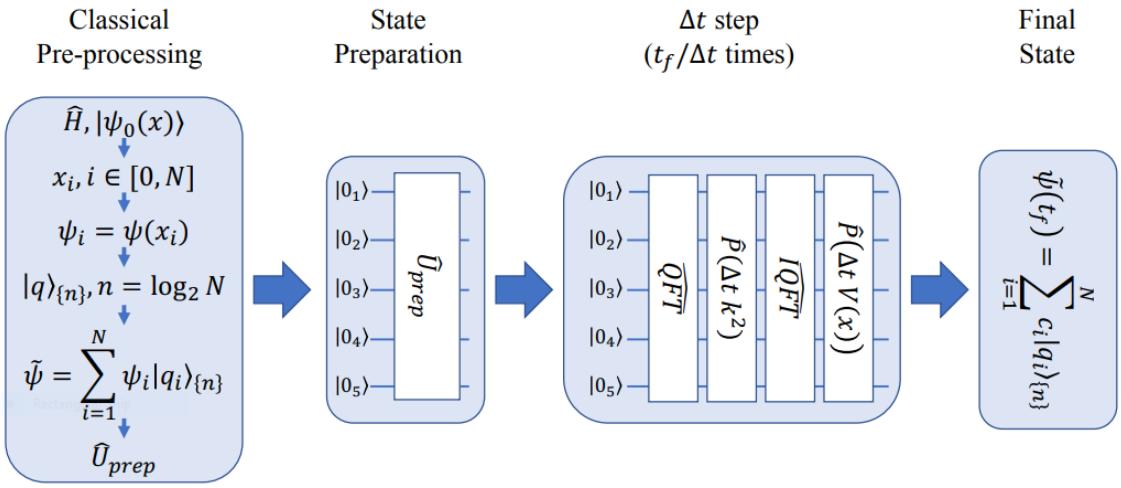

Figure 4.1: These are the steps for the Schrodinger equation's simulation. First of all, prepare the operator U_{prep} classically by converting the wave function into qubits. This operator helps to find the wanted state $|\psi(t_f)\rangle$. In the next step, the operator applied to all zero states in the quantum computer helps to prepare the initial state. In the 3rd step, the operator splitting method is used which has to be repeated by $\triangle t$ time steps. Here $\triangle t$ is the same as ϵ used in the above discussion. In the final step, we get our desired state. This picture is regenerated from [1].

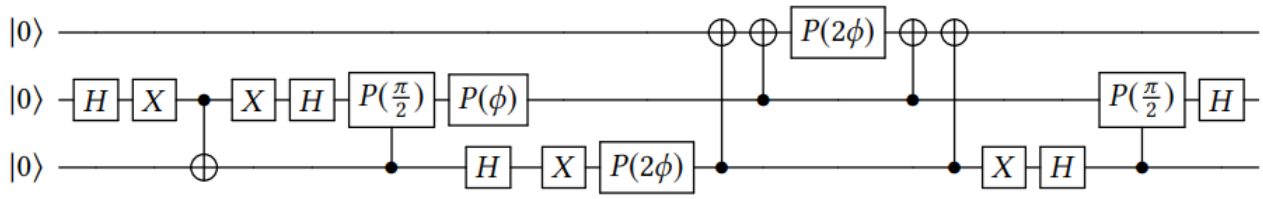

Figure 4.2: This circuit executes Algorithm 1 for 2-qubits. QFT applied to the initial state. Then 1-qubit phase shift applied and 2-qubit phase shift applied by using ancillary qubit q_0. Undo this previously applied QFT by applying $QFT^{\dagger}$ to do the measurement on a computational basis (circuit end). This circuit is rejuvenated from [1].

where the state accomplishes the phase shift in ϵ time step is denoted by ϕ. The propagator $e^{-\iota P^2 t_f}$ has one and two-qubit phase shift gate implementation. As cleared from Ref. [4] that an ancillary qubit called for the two-qubit phase shift operation. This means that the 3-qubits are required for implementation on a quantum computer. To achieve the resultant state on a computational basis, the inverse quantum Fourier transform is applied. The whole process is shown in the Fig. 4.2 .

4.0.2 Algorithm implemented on IBM's 5-qubit computer

We have explained how to implement 2-qubits using the solution to the motion of a single particle. IBMQ's computational results have shown here classically using the Qiskit software. It is Python based software that allows anyone to experiment on the IBMQ quantum computer. This software is divided into four modules, and here in this paper, we are working in the Aer module which handles a specific part of Qiskit. We have used Qiskit as a Python library which proceeds into circuit designing and running it. $QuantumCircuit$ class is used for circuit designing in Qiskit. To simulate it on an IBMQ quantum computer, the simulator used is quantum_simulator. A sample Python code to simulate the single-particle Schrodinger equation on a quantum processor by using the Qiskit library is

```
from qiskit import *
from math import pi
phi=pi/2    #pi, 3*pi/2, 2*pi
qr=QuantumRegister(3)
cr=ClassicalRegister(2)
qc=QuantumCircuit(qr,cr)
%matplotlib inline
qc.u1(pi,qr[1])
qc.h(qr[2])
qc.cx(qr[2],qr[1])
qc.u1(pi,qr[1])
qc.u2((pi/4),pi,qr[2])
qc.cx(qr[2],qr[1])
qc.u1(-pi/4, qr[1])
qc.cx(qr[2],qr[1])
qc.u1((pi/4)+phi,qr[1]) #phi
qc.u2(2*phi, 2*phi, qr[2]) #phase
qc.cx(qr[2], qr[0])
qc.cx(qr[1], qr[0])
qc.u1(2*phi,qr[0])   #phase
```

```
qc.cx(qr[1],qr[0])
qc.cx(qr[2],qr[0])
qc.u2(3*pi/4,pi,qr[2])
qc.cx(qr[2],qr[1])
qc.u1(pi/4, qr[1])
qc.cx(qr[2],qr[1])
qc.u2(0,3*pi/4,qr[1])
qc.measure(qr[1],cr[0])
qc.measure(qr[2],cr[1])

sim=Aer.get_backend('qasm_simulator')
execute(qc,backend=sim)
Result=execute(qc, backend=sim).result()
from qiskit.tools.visualization import plot_histogram
qc.draw(output= 'mpl')
```

The implementation of the above algorithm is shown in Fig. 4.1. The gates U_1 and U_2 are unitary rotation gates performing the operation as given in Eq. (3.18). We have applied a control gate, and we want to apply phase correction if and only if the control qubit is $|1\rangle$. As QFT is applied in superposition, instead of making the circuit twice for state $|0\rangle$ and $|1\rangle$, we use the Hadamard gate.

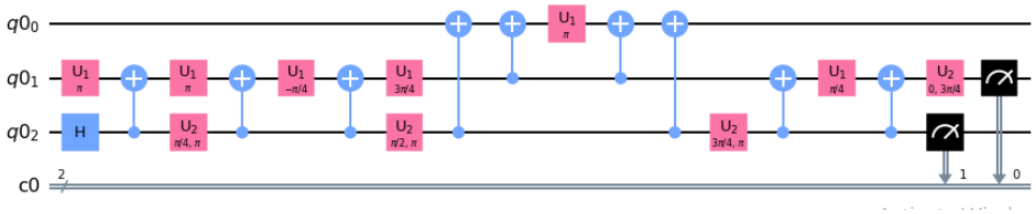

Figure 4.3: It is circuit is for the simulation of the Schrodinger equation on a quantum processor. This circuit is drawn for 2-qubits by using the Qiskit library and work according to Algorithm 1 on the ibmq×4 quantum computer for the special case of $\phi = \pi/2$. The same can be done for other values of ϕ.

The measurement was done for different values (0, $\pi/2$, π, $3\pi/2$, 2π) of ϕ in computational basis for qubit states ($|00\rangle, |01\rangle, |10\rangle, |11\rangle$). As an example the probability of finding a particle in qubit states for $\phi = \pi/2$ is shown in Fig. 4.3. Initial wave function (Π-function) was for $\phi = 0$ with probabilities were {0, 0.499, 0.501, 0}. As value of ϕ increases i.e π, the wave function evolves in times in such a way that Π-function changed to uniform function. For $\phi = \pi$, the probabilities were {0.240, 0.249, 0.243, 0.268}. Then for further increase in value of ϕ i.e from π to 2π, the uniform wave function came back to Π-function. This evolution can be shown from probabilities

{0.119, 0.136, 0.117, 0.628} and {0, 0.490, 0.510, 0} for $\phi = 3\pi/2$ and $\phi = 2\pi$ respectively.

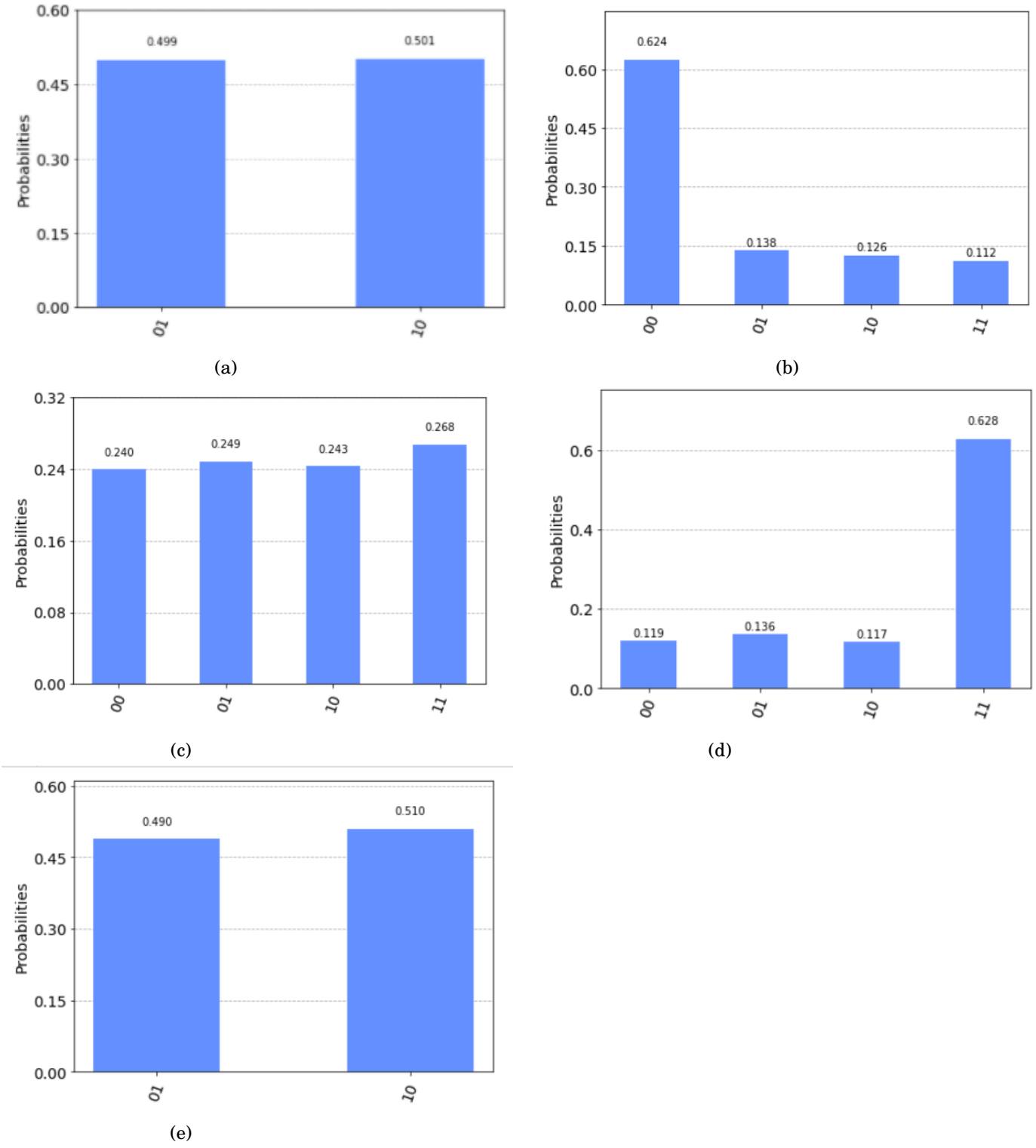

Figure 4.4: Quantum simulator results for 2-qubits while running quantum circuit Figure(4.4): (a) shows the probabilities of qubit state for zero phase rotation i.e $\phi = 0$. This shows that wave function is in initial state. (b) Shows the probabilities of qubit state for $\phi = \pi/2$. (c) Shows the probabilities of qubit state for $\phi = \pi$. (d) Shows the probabilities of qubit state for $\phi = 3\pi/2$. (e) Shows the probabilities of qubit state for $\phi = 2\pi$

4.1 Conclusion

The task of simulation of Schrodinger equation of single free particle ($V(x) = 0$) in one dimension on the quantum processor has been completed. I used a quantum simulation algorithm for 2-qubits that was implemented on a quantum computer to solve the Schrodinger equation. We used Python code to get quantum circuits by using Qiskit (Python-based software). We showed that how wave function (rectangular wave) evolves in time for the particular value of Hamiltonian when this circuit implements algorithm on a quantum computer (ibmq×4). The graphs explained the particle's position in terms of probabilities qubit states for different phase shifts. I showed that how the quantum computer was helpful to solve the physical problem in terms of less time and energy.

References

[1] J. Abhijith, A. Adedoyin, J. Ambrosiano, P. Anisimov, A. Bärtschi, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, et al., *Quantum algorithm implementations for beginners*, arXiv e-prints, (2018), pp. arXiv–1804.

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al., *Quantum supremacy using a programmable superconducting processor*, Nature, 574 (2019), pp. 505–510.

[3] G. Benenti and G. Strini, *Quantum simulation of the single-particle schrödinger equation*, American Journal of Physics, 76 (2008), pp. 657–662.

[4] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, 2002.