ELEMENTS OF A QUANTUM VIRTUAL MACHINE AND GPU
IMPLEMENTATION OF BLACK-SHOLES EQUATION

by

AKASH KAVITKAR

B.E., University of Pune, 2014

A thesis submitted to the Graduate Faculty of the

University of Colorado Colorado Springs

In partial fulfillment of

the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

2018
This thesis for the Master of Science degree by

Akash Kavitkar

has been approved for the

Department of Electrical and Computer Engineering

by

Carlos De Paz Araujo, Chair

T.S. Kalkur

Heather Song

Date: May 7th, 2018
Kavitkar, Akash (M.S., Electrical Engineering)

Elements of a Quantum Virtual Machine and GPU Implementation of Black-Sholes

Thesis directed by Professor Carlos De Paz Araujo

ABSTRACT

The Quantum Virtual Machine is a wavefunction simulation of unitary evolution with classical control flow and shared quantum classical memory. The quantum gate can perform all necessary basic operations of the quantum computer. The electron spin of individual electrons in silicon is used as the basic storage unit ("quantum bits"). The scope of the thesis is restricted to understanding Josephson Junctions by examining the differential equations and tilted washboard potential model, as such, they possess discrete levels in their effective potential wells, and would escape from those wells with the appearance of a finite junction voltage via a macroscopic quantum tunneling process and solve some basic quantum problem. Graphics Processing Unit is used to illustrate parallel computing to solve Black-Sholes equation as an example. Calculations are achieved by a GPU (Graphics Processing Unit), to demonstrate computational abilities for numerical operation. We used NVidia CUDA to perform the computations. We also introduce the path Integral Formulation along with detailed mathematical description of integrating it in the quantum computation.
ACKNOWLEDGEMENTS

I want to appreciate and be to the people who were there for me and supported me during this beautiful journey.

I would like to thank Dr. Carlos for his unparalleled support and guidance throughout the journey, it would not have been possible without him. It was truly a great experience working with him. I would like to thank all the professors who guided me thought-out my master’s degree program. I would like to thank Greg Jones for supporting me in my endeavors. I was a pleasure to work with him, I have learned so much in such a short amount of time at Symmetrix Corps.

I would also like to extend my gratitude towards my friends in the United States Ramaprasad, Baburao, Shahrulkh, Mrunal, Aamir, Surabhi, Dharini and all my friends at Symmetrix for their moral support and who have always been there with me.

Finally, I would like to thank my loving parents and my little sister for believing in me and supporting me in this adventure.
# TABLE OF CONTENTS

## CHAPTER

1 INTRODUCTION 1

1.1 Superconductivity ................................................................. 2

1.2 Cooper Pairs ........................................................................... 3

2 JOSEPHSON JUNCTION 5

2.1 Josephson Junction ................................................................. 5

2.2 Non-Linear Josephson Inductance ........................................... 6

2.3 Three basic types of Josephson Qubits ................................... 7

2.4 Quantum Logic gate ............................................................... 10

2.5 Josephson Effect ................................................................. 11

2.6 Single Josephson junction ..................................................... 14

2.7 Single Josephson Junction device Model ............................... 15

2.8 Escape from a Well ............................................................. 18

2.9 The simple pendulum Analogy ........................................... 19

3 JOSEPHSON JUNCTION WITH PATH INTEGRAL MODEL 20

3.1 Path Integral Model ............................................................ 20
3.2 Single-Josephson junction Lagrangian and Hamiltonian.................24

3.3 Short time expansion......................................................................27

4 PARALLELIZATION AND GPU PROGRAMMING 28

4.1 Background......................................................................................28

4.2 Multiprocessors, work groups, Threads and Global sizes...............29

4.3 Memory ............................................................................................29

4.4 Speedup in parallel programming......................................................31

4.5 Challenges in GPU Implementation................................................31

4.6 Kernel runtime limitation ..................................................................32

5 EUROPEAN OPTION PRICING FOR BLACK-SHOLES EQUATION IN CUDA 33

5.1 Theory................................................................................................33

5.2 Path integral model of European option pricing of Black-Sholes.......35

6 RESULTS 37

6.1 Implementation of Black-Sholes in CUDA .......................................37

6.2 Implementation of RSCJ Josephson Junction in MATLAB ..........43

7 CONCLUSIONS 48
8 FUTURE WORK

REFERENCES
LIST OF FIGURES

FIGURE

1.1.1 Basic elements of a Quantum Virtual Machine.................................1

1.2.1 The Cooper pairs ..............................................................................4

2.1.1 Diagram of single Josephson Junction.............................................5

2.3.1 Flux qubit circuit..............................................................................7

2.3.2 Schematic potential energy for Flux circuit.................................8

2.3.3 Phase qubit circuit...........................................................................8

2.3.4 Washboard potential of current biased junction............................9

2.5.1 Schematic representation of a Josephson Junction.......................12

2.7.1 Resistive and capacitive shunted model......................................15

2.7.2 Washboard Potential for Josephson effect.................................17

2.7.3 Washboard Potential for Josephson effect.................................18

2.9.1 A simple pendulum......................................................................19

4.1.1 GPU and CPU..............................................................................29
4.3.1 Memory hierarchy

6.1.1 Output of the CUDA code

6.2.1 Phase verse time graph with Washboard potential

6.2.2 Phase verse time graph with Washboard potential zoomed

6.2.3 Graph of phase verse time for three Josephson junctions
CHAPTER 1

INTRODUCTION

Most proposals for implementing a quantum computer have been based on qubits constructed from microscopic degrees of freedom: spin of either electrons or nuclei, transition dipoles of either atoms. The main challenge of these implementations is enhancing the inter-qubit coupling to the level required for fast gate operations without introducing decoherence from parasitic environmental modes due to thermalization and noise.

Just as a classical digital computer uses the "bit" with the values of either zero or one as the basic unit for all the calculation processes, a quantum computer, too, needs a basic storage unit: the quantum bit. The difference is that the quantum bit is not limited to two states (zero and one) but can exist in multiple states at once and is therefore much more complex in its implementation than a simple digital system. The electron's direction of rotation corresponds to the zero and one values of the digital bit, but in its exact quantum state the electron can hold more information than just zero or one.

**Figure 1.1.1** Basic elements of a Quantum Virtual Machine
Josephson Junctions are great contender for the development of quantum bits (qubits) for a quantum computer. Josephson junctions are ideal because of low power dissipation. In this thesis we focus on radically different Elements of “Virtual quantum machine.” Qubits are constructed from collective electrodynamic modes of macroscopic electrical elements (Josephson Junction), rather than microscopic degrees of freedom. Single Josephson Junction model is implemented in MATLAB ode23s solver. Representation of the Josephson junction as a wave function in Path Integral form to demonstrate that the short comes of the Hamiltonian model could be overcome by Path Integral Form which is our way to find a solution of the problem with a Path Integral. We Also Introduce parallel programming and cross talk between the threads with the entangled qubits and executed in GPU environment and its uses in this thesis. Implement the Black-Sholes model in CUDA without the quantum finance approach.

1.1 Superconductivity

Here we expand the material we have used in this thesis that will help the reader with many references and basic concepts related to Josephson junction and its effects. In 1911, the Dutch physicist Heike Kamerlingh Onnes, while studying the properties of matter at very low temperature observed that the electrical resistance of some metals goes to zero below some material specific critical temperature $T_c$. That was the observation of the phenomenon called superconductivity. Although superconductivity is not in the scope of this thesis, but it will be good to have some background and understanding of basic principles as it plays an important role in understanding the physics related to the thesis.

We can characterize superconductivity in manly two ways. First observed property of a superconductor is that there is no loss of current or magnetic field produced by the current in superconductor. Second property is called the Meissner effect, where in
the superconductor expels all magnetic fields as they are cooled below the critical temperature $T_c$. This intern indicates that superconductivity can be destroyed by applying a strong magnetic field. When placed in a critical magnetic field $H_c$ the superconductivity disappears. There are two types of superconductors, Type-I materials remain superconductive only below certain week magnetic field $H_c$ until it breaks down discontinuously. Type-II materials tolerate local magnetic field penetration, which enables them to have superconducting properties in the presence of high magnetic fields. This behavior is explained by the existence of a mixed state, it has a lower critical magnetic field $H_{c1}$ and higher critical magnetic field $H_{c2}$, where the superconductor breaks down continuously. Type-II superconductors will keep the whole magnetic field out until $H_{c1}$. After $H_{c1}$ vortices start to appear. Also known as magnetic flux quantum that penetrates the superconductor and is mathematically described by equation 1.1. where $e$ is the electron charge, $h$ is Planck’s constant and $c$ is the speed of light.

$$\phi = \frac{hc}{2e}$$  

(1.1)

1.2 Cooper pairs:

Conventional physics does not effectively explain the superconducting state. In 1957 John Bardeen, Leon Cooper and John Schrieffer established the microscopic theory of superconductivity. BCS theory explains that the electrons group into pairs due to the interaction with vibrations of the lattice, thus forming “Cooper pairs” which move in the solid without friction. The Cooper pairs consists of two electron pairs, one in state $-k \downarrow$ and the other in $-k \uparrow$. 
The BCS theory explained the superconducting current as a super fluid of Cooper pairs which are a pair of electrons interacting through the exchange of phonons. Cooper pairs are formed by an attractive force between electrons from the exchange of phonons. The energy of a phonon is usually less than 0.1 eV. Figure from [1]

**Figure 1.2.1** Cooper pairs [1]
CHAPTER 2

JOSEPHSON JUNCTION

2.1 Josephson Junction

A Josephson Junction is a type of electric circuit which is made by sandwiching a thin layer of non-superconducting material between two layers of superconducting material. The circuit can switch at high speeds when operated at temperatures moving toward absolute zero.

![Diagram of single Josephson Junction](image)

**Figure 2.1.1** Diagram of single Josephson Junction. Insulator is weak link between the two superconductors [2].

The non-superconducting barrier separating the two superconductors must be very thin in the order of 30 angstroms and less. When a certain critical current is reached, a supercurrent flows through the barrier, electron pairs can tunnel across with zero resistance. The devices are named after Brian Josephson, who anticipated in 1962 that sets of superconducting electrons could "tunnel" directly through the non-superconducting barrier starting with one superconductor then onto the next.
2.2 Non-Linear Josephson Inductance

The Josephson effect describes the supercurrent $I_j$ that flows through the junction according to the equations [3]

\[ I_j = I_c \sin \delta \]  \hspace{1cm} (2.2.1)

\[ V = \frac{\phi_0}{2\pi} \frac{d\delta}{dt} \]  \hspace{1cm} (2.2.2)

Where $\phi_0 = h/2e$ is the superconducting flux quantum, $I_c$ is the critical current and $\delta$ is the superconducting phase difference across the junction. The behavior of equation 2.10 can be found out if we differentiate 2.10 and replace $d\delta/dt$ with $V$. Therefore, we get

\[ \frac{dI_j}{dt} = I_0 \cos \delta \frac{2\pi}{\phi_0} V \]  \hspace{1cm} (2.2.3)

This equation is an inductor. By definition $V = L_j dI_j/dt$ The Josephson inductance $L_j$ is given as

\[ L_j = \frac{\phi_0}{2\pi I_c \cos \delta} \]  \hspace{1cm} (2.2.4)

The Stored energy of the system is given by

\[ U_j = \int \int I_j V dt \]

\[ U_j = \int \int I_c \sin \delta \frac{\phi_0}{2\pi} \frac{d\delta}{dt} dt \]

\[ U_j = -\frac{I_c \phi_0}{2\pi} \cos \delta \]  \hspace{1cm} (2.2.5)
A Hamiltonian describes the classical and quantum behavior of a circuit. Hamiltonian corresponds to Total Energy of the system. It is denoted by $\hat{H}$. It depends on the exact circuit configuration. In general, the Hamiltonian of the Josephson effect is $H_J = U_J$. 

### 2.3 Three basic types of Josephson qubits

There are three fundamental types of qubits which consist of one Josephson element. A Josephson qubit can be realized as a nonlinear oscillator by modeled the Josephson inductance and its junction capacitance. The operating space must be restricted to two lowest states. The frequency $\omega_{01}$ that changes between the states 0 and 1 should be different from the frequency $\omega_{12}$ that changes between states 1 and 2. [3][4]

Phase, Flux, and Charge qubits are the three type of qubits which we discuss. We use the Phase qubit.

**a.** Flux qubit

![Figure 2.3.1 Flux qubit circuit [3]](image)

The circuit for the flux-qubit circuit is given in the figure. The two sides of the junction with capacitance C are connected by a superconducting loop with inductance L. Flux is imposed through the loop by a coil. Its Hamiltonian is given by
\[ H = \frac{\dot{Q}^2}{2C} - \frac{I_c\phi_0}{2\pi} \cos \dot{\delta} + \frac{1}{2L} \left( \frac{\phi - \phi_0}{2\pi} \right)^2 \] (2.3.1)

Figure 2.3.2 Schematic potential energy for Flux circuit [5].

The charge \( \dot{Q} \) on the capacitance \( C \) obey \([\dot{\delta}, \dot{Q}] = 2ei \). The circuit is immune to effect of offset charge and offers reduced charge fluctuation effects.

b. Phase qubit.

Figure 2.3.3 Phase qubit circuit[3].

The second quantum circuit biases the junction with a DC-current source. The inductance obtains non-linearly applying junction current \( I \) close to the critical current. The circuit is immune to offset charge and offers reduced charge fluctuation effects like the flux qubit. The Hamiltonian can be realized as
\[ H = \frac{\hat{Q}^2}{2C} - \frac{I_c \phi_0}{2\pi} \cos \hat{\delta} - \frac{I \phi_0}{2\pi} \hat{\delta} \] (2.3.2)

Note that the phase and flux qubit Hamiltonians are equivalent for \( L \to \infty \) and \( I = \Phi/L \), which resembles to a current bias created from an inductor with infinite impedance. The potential \( \hat{\delta} \) in the representation is shown in figure below. It is a Tilted Washboard Potential with the tilt being the ration of \( I/I_c \).

![Figure 2.3.4](image)

**Figure 2.3.4** Washboard potential of current biased junction [5].

Total potential energy of the circuit can be approximated as

\[ U(\delta) = \phi_0(I_c - I)(\delta - \pi/2) - \frac{I_c \phi_0}{6}(\delta - \pi/2)^3 \] (2.3.3)

Energy levels are found in the well. The first two levels can be used for qubit states. The qubit circuit is based on the possibility that states in the potential can tunnel through the potential. After tunneling happens, the particle representing the phase travels down the tilted washboard representing qubit state change. The behavior of each qubit is governed by the laws of quantum mechanics, enabling qubits to be in a “superposition”
state – that is, both a 0 and a 1 at the same time, until an outside event causes it to “collapse” into either a 0 or a 1.

2.4 Quantum logic gate

Quantum logic gate is a small quantum circuit operating on small number of qubits. It is the building block of quantum circuit. Quantum gates operate on one or two qubits and are represented by matrices. Quantum gates can be represented by $2^n \times 2^n$ unitary matrix, where $n$ is the number of qubits the gate is operating on.

Therefore, gate which acts on $n$ qubits will have $2^n \times 2^n$ unitary matrix and the number of qubits at input and output should be equal. A given quantum state can be found by multiplying the vector which represents the state by unitary matrix of the gate.

A single qubit vector can be represented by

$$v_0 |0\rangle + v_1 |1\rangle \rightarrow \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \quad (2.4.1)$$

representation of two qubits is

$$v_{00} |00\rangle + v_{01} |01\rangle + v_{10} |10\rangle + v_{11} |11\rangle \rightarrow \begin{bmatrix} v_{00} \\ v_{01} \\ v_{10} \\ v_{11} \end{bmatrix} \quad (2.4.2)$$

Where $|ab\rangle$ is the basis vector representing a state, $|a\rangle$ is the first qubit and $|b\rangle$ is second qubit.
2.5 **Josephson Effect**

The Josephson effect is an example of a macroscopic quantum phenomenon. It is named after the British physicist Brian David Josephson, who predicted in 1962 the mathematical relationships for the current and voltage across the weak link[6][7]. The paper predicted that a current can flow through a circuit consisting of two superconducting metals, separated by a thin insulating film.

The DC Josephson effect had been found in tests before 1962, yet had been ascribed to "super-shorts" or breaks in the protecting hindrance prompting the immediate conduction of electrons between the superconductors. The first paper to claim the discovery of Josephson's effect, and to make the requisite experimental checks, was that of Philip Anderson and John Rowell. These authors were awarded patents on the effects that were never enforced, but never challenged. [8]

Prior to Josephson's forecast, it was just realized that typical (i.e. non-superconducting) electrons can move through an insulating barrier, by methods of quantum tunneling. Josephson was the first to foresee the tunneling of superconducting Cooper sets. For this work, Josephson got the Nobel Prize in Physics in 1973.

The active device in superconductor electronics is a junction between two superconductors which is weak enough to allow only a slight overlap of the electron pair wave functions of the two superconductors. Under this condition, electron pairs can pass from one superconductor to the other even with no applied voltage. Numerous ways of forming such weak links have been explored for both metallic and oxide superconductors. These include metal or semiconductor links, grain boundaries, very narrow constrictions, damaged regions, and, most prominently, insulating tunnel barriers. To date it has not been possible to demonstrate a tunnel junction in the technologically interesting high-temperature superconductors, but tunnel junctions play a prominent role in LTS
electronics. For this reason, and that the tunnel junction offers the best vehicle for elaboration of many of the important characteristics of Josephson junctions, we will focus most of our analyses on these devices.

This effect can be derived as follows:

Figure 2.5.1 Schematic representation of a Josephson Junction[8].

The Josephson effect is the physical phenomena describing what occurs when we have quantum tunneling between superconductors. This effect can be derived as follows: Ref. [9]

Let $\Psi_1$ be the probability amplitude for electron pairs to be on one side of a junction while $\Psi_2$ is the probability amplitude for electron pairs to be on the other side of the junction. Then the time-dependent Schrödinger equation.

$$ i \hbar \frac{\partial \Psi_1}{\partial t} = \hbar \Psi_1 T \quad i \hbar \frac{\partial \Psi_2}{\partial t} = \hbar \Psi_2 T $$

(2.5.1)

where $\hbar T$ represents the effect of the electron-pair coupling or transfer interaction across the insulator. $T$ is a measure of the leakage of $\Psi_1$ into region 2 and of $\Psi_2$ into region 1. The wave function in each region can be described as $\Psi_1 = n_1^{1/2} e^{i\theta_1}$ and $\Psi_2 = n_2^{1/2} e^{i\theta_2}$. Then,
\[
\frac{\partial \Psi_1}{\partial t} = \frac{1}{2} n_1^{-1/2} e^{i\theta_1} \frac{\partial n_1}{\partial t} + i\Psi_1 \frac{\partial \theta_1}{\partial t} = -iT\Psi_2
\] (2.5.2)

\[
\frac{\partial \Psi_2}{\partial t} = \frac{1}{2} n_2^{-1/2} e^{i\theta_2} \frac{\partial n_2}{\partial t} + i\Psi_2 \frac{\partial \theta_2}{\partial t} = -iT\Psi_1
\] (2.5.3)

If we multiply first of the two equations by \( n_1^{1/2} e^{i\theta_1} \) and the second by \( n_2^{1/2} e^{i\theta_2} \), we obtain, with \( \delta = \theta_2 - \theta_1 \)

\[
\frac{1}{2} \frac{\partial n_1}{\partial t} + in_1 \frac{\partial \theta_1}{\partial t} = -iT(n_1 n_2)^{1/2} e^{i\delta}
\] (2.5.4)

\[
\frac{1}{2} \frac{\partial n_2}{\partial t} + in_2 \frac{\partial \theta_2}{\partial t} = -iT(n_1 n_2)^{1/2} e^{-i\delta}
\] (2.5.5)

We now equate the real and imaginary parts on the left and the right side of these equations and attain

\[
\frac{\partial n_1}{\partial t} = 2T(n_1 n_2)^{1/2} \sin \delta
\] (2.5.6)

\[
\frac{\partial n_2}{\partial t} = -2T(n_1 n_2)^{1/2} \sin \delta
\] (2.5.7)

\[
\frac{\partial \theta_1}{\partial t} = -T\left(\frac{n_1}{n_2}\right)^{1/2} \cos \delta
\] (2.5.8)

\[
\frac{\partial \theta_2}{\partial t} = -T\left(\frac{n_2}{n_1}\right)^{1/2} \cos \delta
\] (2.5.9)

If superconductors in region 1 and 2 are identical, then \( n_1 = n_2 \) and Eq. (2.5.9) imply that

\[
\frac{\partial \theta_2}{\partial t} = \frac{\partial \theta_1}{\partial t}
\] (2.5.10)

\[
\frac{\partial(\theta_2 - \theta_1)}{\partial t} = 0
\] (2.5.11)
while Eq. (2.5.6) imply that

\[
\frac{\partial n_2}{\partial t} = -\frac{\partial n_1}{\partial t} \quad (2.5.12)
\]

The current flow is proportional to \( \frac{\partial n_2}{\partial t} \) or, \(-\frac{\partial n_1}{\partial t}\). From Eq. (2.6.6) we can conclude that the current \( I_J \) is dependent only on the phase difference \( \delta \), as

\[
I_J = I_c \sin \delta \quad (2.5.13)
\]

where \( I_c \) is proportional to the transfer interaction \( T \).

2.6 Single Josephson junction

A system in superconducting state can be described by a wave function \([10][11]\).

\[
\Psi = |\Psi| e^{i\varphi} \quad (2.6.1)
\]

When two superconductors are placed closely and separated by a thin barrier, cooper pairs can flow through the barrier, which makes the two systems interact with each other. In terms of phase difference.

\[
\gamma = \varphi_a - \varphi_b - \left( \frac{2\pi}{\phi_0} \right) \int_a^b A ds \quad (2.6.2)
\]

Where \( \phi_0 = \hbar c / 2e \) is the quantum flux, with \( \hbar \) the Planks constant, \( c \) the light velocity in vacuum and \( e \) as the electron charge.
2.7 Single Josephson Junction device Model

Keeping in mind that the real Josephson junction is not perfect, we use the classical approach for a simple equivalent circuit of Josephson junction the resistive capacitive shunted junction model (RCSJ) [12].

![Resistive and capacitive shunted model of a Josephson junction (RCSJ)](image)

**Figure 2.7.1** Resistive and capacitive shunted model of a Josephson junction (RCSJ) [12].

This circuit has the essential elements of a real Josephson junction device, there are three elements in parallel: shunt resistor $R$, shunt capacitance $C$, and the ideal Josephson junction $JJ$. Total current applied across the circuit is $I$.

The voltage is proportional to the phase shift of the wave function as the current flows through the circuit, exactly $V = \frac{h}{2e} \frac{d\psi}{dt}$.

If a finite frequency bias current is applied, the total current through the Josephson junction can now split three ways in general. By Kirchhoff's current law we get the current equation.

$$\frac{hC}{2e} \frac{d^2\psi}{dt^2} + \frac{h}{2eR} \frac{d\psi}{dt} + I_c \sin \psi = I \quad (2.7.1)$$
Let \( w_{j0} = \sqrt{\frac{2eI_c}{\hbar C}} \) be the plasma frequency of a zero biased Josephson Junction.

Where \( \alpha = \frac{\hbar w_{j0}}{2eI_c R} \) is a normalized loss coefficient and \( \eta = \frac{I}{I_c} \) is the normalized current.

Therefore, we can simplify equation 2.7.1 as follows:

\[
\frac{d^2\psi}{dt^2} + \alpha \frac{d\psi}{dt} + \sin \psi = \eta
\]  (2.7.2)

**Tilted Washboard Potential**

The tilted washboard model exemplifies the phase of the Josephson junction as well as the voltage and current. It is a mechanical analogy to visualize the behavior of Josephson junction, we image a ball shaped mass resting on a corrugated one-dimensional surface of a washboard. The ball is subjected to various degrees of tilt and wiggling. One can think of Johnson noise in the resistor as being analogous to Brownian motion of the particle in a viscous fluid that provides the drag.

In this analogue the current is the slope of the washboard and the velocity of the mass over the washboard is interpreted as the voltage over the junction. Starting from a horizontal washboard \((I = 0)\) imagine tilting it to one side as the dc current is applied.

The washboard is given a slight tilt but the mass will still be at rest in the small valleys of the corrugated washboard. Thus, we have an inclined plane with zero velocity for the mass is analogous to the superconductivity of the junctions when the current is below the critical current \( I_c \).
When a certain angle is reached where tilt is increased the phase point, or the mass is unstable and will begin to run, putting the junction in to the finite-voltage state. There remain no potentials where the mass can be at rest and it will thus have a velocity, representing the voltage appearing in the junction.

The total potential energy of a junction, when an additional bias current is supplied, is Ref[13]

\[ U = E_J \{(1 - \cos \psi) - \eta \psi\} \]  

(2.7.3)

In this form it is seen that the phase dynamics can be viewed in terms of a mass whose co-ordinates is \( \psi \) indicated in figure 2.7.3.

**Figure 2.7.2** Washboard Potential for Josephson effect shows the current as the movement of a mass over the washboard. Washboard potential with zero bias and non-zero bias [14].
2.8 Escape from a Well

$\Delta U$ is the barrier height for the washboard potential, shown in figure 2.4, is controlled by a bias current $\eta$ and is described as in equation 2.7.3.

$$\Delta U = 2E_f \{\sqrt{(1-\eta^2)} - \eta \cos^{-1} \eta\}$$  \hspace{1cm} (2.8.1)

If noise is present, then the mass can hop out of the well and bounce down the washboard. For thermal activation, the escape rate at temperature $T$ is given by

$$T(t) = f \exp\left(\frac{-\Delta U}{k_B T}\right)$$  \hspace{1cm} (2.8.2)

As $\frac{d\phi}{dt} = 2eV/\hbar$ this bouncing motion is achieved by an oscillating but always positive junction voltage.

Figure 2.7.3. Part of the washboard potential model showing the barrier escape potential [15].
2.9 The simple pendulum Analogy.

The pendulum approximates like a simple harmonic oscillator for small swings, equation that governs the motion of a driven pendulum, as depicted in Fig.2.7.4. For the pendulum, the equation of motion is

\[ l \frac{d^2 \varphi}{dt^2} + \gamma \frac{d\varphi}{dt} + mgl \sin \varphi = T \] (2.9.1)

Where \( m \) is mass of the bob, \( l \) is the length of suspension, \( I \) is rotational moment of inertia, \( \gamma \) is the damping factor and \( T \) is the applied torque. The potential energy of the system for displacement angle \( \varphi \) and applied torque is

\[ U = mgl(1 - \cos \varphi) + T\varphi \] (2.9.2)

Therefore, the single junction (RCSJ) model can be have an analogy of a damped pendulum.
## 3.1 Path Integral Model

While studying how to cross over from classical model for Josephson junction to the quantum model of Josephson junction, Richard Feynman proposed a method deriving Josephson’s equations in which supercurrent is predicted as a result of phase interface between two weakly coupled quantum systems [16] [17].

The Feynman’s derivation of the Josephson junction is stated explicitly as possible so that the model can be expanded further to include applications such as Black-Sholes equation.

1. The individual superconductors would be an energy eigen states and will be represented by wavefunctions.

\[ \psi_1(t) = \psi_1(0)e^{i\phi_1} \]  
\[ \psi_2(t) = \psi_2(0)e^{i\phi_2} \]

2. When we sandwich the superconductor with a thin non-conducting film separating them. The resulting wavefunction of the system is a superposition of the individual wavefunctions due to the coupling is given by

\[ \psi = \alpha \psi_1 + \beta \psi_2 \]

3. Superconductors exchange particles. The number density of the particles is:

\[ n_1 = \|\alpha\|^2, n_2 = \|\beta\|^2, n_{total} = \|\alpha\|^2 + \|\beta\|^2 = 1 \]
4. The wavefunction of the total system can be described as:

\[ \psi(t) = \sqrt{n_1} e^{i\phi} \psi_1(0) + \sqrt{n_2} e^{i\phi} \psi_2(0) \]  

(3.1.5)

Where \( n_1, n_2, \phi_1, \phi_2 \) are time dependent.

5. Coupling of two semiconductors is time independent and symmetric.

\[ \partial_t \hat{K} = 0 \quad \text{where} \quad \langle \psi_1 | \hat{K} | \psi_2 \rangle = \langle \psi_2 | \hat{K} | \psi_1 \rangle \]  

(3.1.6)

Where \( \hat{K} \) is the interaction energy of the coupled superconductors.

6. The two superconducting materials have nearly equal number of particles, also the current between them is comparatively small to bring a significant change the number of electrons in the system i.e. \( n_1 \approx n_2 \).

7. The partial current density function from one superconductor to other is written in terms of \( \alpha \) and \( \beta \).

\[ \alpha = \sqrt{n_1} e^{i\phi} \Rightarrow \dot{\alpha} = \frac{1}{2} (n_1)^{-1/2} \dot{n_1} e^{i\phi} + \dot{\phi} \sqrt{n_1} e^{i\phi} \]  

(3.1.7)

\[ \alpha(\dot{\alpha}) = (\alpha \dot{\alpha}) = \frac{1}{2} \dot{n_1} - i \dot{\phi} n_1 \]  

(3.1.8)

\[ \alpha(\dot{\alpha}) + \alpha(\dot{\alpha}) = \frac{1}{2} \dot{n_1} + \frac{1}{2} \dot{n_1} = \dot{n_1} \]  

(3.1.9)

\[ \dot{n_1} = \alpha(\dot{\alpha}) + \alpha(\dot{\alpha}) \]  

\[ \dot{n_2} = \beta(\dot{\beta}) + \beta(\dot{\beta}) \]  

(3.1.10)
8. From equations 3.1.4 and 3.1.5, total number of particles must be conserved in a system. Thus, the currents must be equal in magnitude and opposite in sign. Schrodinger’s equation.

\[ i \hbar \dot{n}_1 = E_i \| \alpha \|^2 + K \alpha \beta - E_i \| \alpha \|^2 - K \alpha \beta = K(\alpha \beta - \alpha \beta) \quad (3.1.11) \]

\[ \dot{n}_1 = -i \frac{K}{\hbar} \sqrt{n_{12}} (e^{i(\phi_2 - \phi_1)} - e^{i(\phi_2 - \phi_1)}) \quad (3.1.12) \]

\[ \dot{n}_1 = 2 \frac{K}{\hbar} \sqrt{n_{12}} \sin(\phi_2 - \phi_1) \quad (3.1.13) \]

Assuming the superconductors form Cooper pairs with charge -2e, \( I_c \) be the supercurrent in the Josephson junction must be -2e times the number of Cooper pairs crossing though to the second superconductor per second:

\[ I_s = -2e \dot{n}_2 = 2e \dot{n}_2 = 4e \frac{K}{\hbar} \sqrt{n_{12}} \sin(\phi_2 - \phi_1) \quad (3.1.14) \]

\[ I_s = I_0 \sin(\gamma) \quad (3.1.15) \]

Where \( \gamma \) is the phase difference between the two superconductors and \( I_0 \) is the critical current. Where \( \gamma = \phi_2 - \phi_1 \) and \( I_0 = 4e \frac{K}{\hbar} n_i \).
Josephson effect also gives relation between derivative of phase difference to the voltage difference across the system.

From equation 3.1.4 we describe $\dot{\phi}_1$ being a function of $\alpha$ as:

$$\alpha(\dot{\alpha}) = \frac{1}{2} \dot{n} + i \dot{\phi}_1 n_1$$  \hspace{1cm} (3.1.16)

$$\alpha(\dot{\alpha}) = \frac{1}{2} \dot{n} - i \dot{\phi}_1 n_1$$  \hspace{1cm} (3.1.17)

$$i\|\alpha\|^2 \dot{\phi}_1 = i n_1 \dot{\phi}_1 = \frac{1}{2} (\alpha \dot{\alpha} - \alpha(\dot{\alpha})) \Rightarrow \dot{\phi}_1 = -i \frac{\alpha \dot{\alpha} - \alpha(\dot{\alpha})}{2\|\alpha\|^2}$$  \hspace{1cm} (3.1.18)

Using Schrodinger’s equation, a similar equation can be described as $\dot{\phi}_2$ being a function of $\beta$

$$\dot{\alpha} = -i \frac{\hbar}{(E_1, \alpha + K \beta)}$$  \hspace{1cm} (3.1.19)

$$\dot{\beta} = -i \frac{\hbar}{(K \alpha + E_2 \beta)}$$  \hspace{1cm} (3.1.20)

$$\dot{\phi}_1 = \frac{i}{2\hbar\|\alpha\|^2} (\alpha(-iE_1\alpha - iK\beta) - \alpha(iE_1\alpha + iK\beta))$$  \hspace{1cm} (3.1.21)

$$\hbar \dot{\phi}_1 = -E_1 - \frac{1}{2\|\alpha\|^2} K(\alpha \beta + \alpha \beta)$$  \hspace{1cm} (3.1.22)

$$\hbar \dot{\phi}_1 = -E_1 - \frac{1}{2n_1} K \sqrt{n_1 n_2} (e^{i(\dot{\phi}_1 - \phi_1)} + e^{i(\dot{\phi}_1 - \phi_1)})$$  \hspace{1cm} (3.1.23)
\[ \hbar \dot{\phi}_1 = -E_1 - K \frac{\sqrt{n_1 n_2}}{n_1} \cos(\gamma) \]  

Similarly, we can find \( \dot{\phi}_2 \) when we replace \( E_1 \) with \( E_2 \).

\[ \hbar \dot{\gamma} = -E_2 - K \cos(\gamma) + E_1 + K \cos(\gamma) = E_1 - E_2 \]  

Energy difference between the two superconductors is proportional to voltage drop across the system which can be given by:

\[ qV = -2eV = E_2 - E_1 \]  

Therefore, we get Josephson junction equation for voltage in relation with flux quantum \( \phi_0 \)

\[ V = \frac{\phi_0}{2\pi} \dot{\gamma} \]  

\[ \phi_0 = \frac{\hbar}{2e} \]  

### 3.2 Single-Josephson junction Lagrangian and Hamiltonian

Lagrangian is a function that describes the state of a dynamic system in terms of position coordinates and their time derivatives and that is equal to the difference between the potential energy and kinetic energy. Lagrangian, a function which summarizes the dynamics of the entire system [16][17].

\[ L = T - V \]  

Where \( T \) is the total kinetic energy of the system and \( V \) is the total potential energy of the system.
This paper considers junctions in which we analyze conventional conductivity and secondary quantum effects such as tunneling and interference in the Josephson junction.

The equation of motion gives the time development and the dynamics of the system. Combining the equation (2.2.11) with RCSJ model we get the following current equation:

\[ I = I_0 \sin(\gamma) + C_j \left( \frac{\phi_0}{2\pi} \right) \dot{\gamma} \]  

(3.2.2)

The goal is to find the Lagrangian equivalent with this equation. The Lagrangian can be formulated by knowing the generalized coordinate and generalized velocity as

\[ L = \left( \frac{\phi_0}{2\pi} \right) \left[ \frac{1}{2} C_j \left( \frac{\phi_0}{2\pi} \right) (\dot{\gamma})^2 + I_0 \cos(\gamma) + I \right] \]  

(3.2.3)

We check for consistency so that Lagrangian satisfies the Euler-Lagrange condition. It can be shown that the equation of motion is recovered from the Lagrangian.

\[ \frac{\partial L}{\partial \dot{\gamma}} = \left( \frac{\phi_0}{2\pi} \right) \left[ -I_0 \sin(\gamma) + I \right] \]  

(3.2.4)

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\gamma}} \right) = \left( \frac{\phi_0}{2\pi} \right)^2 C_j \ddot{\gamma} \]  

(3.2.5)

Which corresponds to

\[ C_j \left( \frac{\phi_0}{2\pi} \right) \ddot{\gamma} = -I_0 \sin(\gamma) + I \]  

(3.2.6)

Therefore, the Lagrangian recovered the correct equation of motion using the Euler-Lagrange equation.
Partial derivative of the Lagrangian transformation is the momentum $P$. 

$$p = \frac{\partial L}{\partial \dot{\gamma}} = \left(\frac{\phi_0}{2\pi}\right)^2 C_j \dot{\gamma}$$  \hspace{0.2cm} (3.2.7)\

The Hamiltonian is calculated using the definition of Legendre transformation of the Lagrange. 

$$H = \dot{\gamma} p - L = \left(\frac{\phi_0}{2\pi}\right) \left[\frac{1}{2} C_j \left(\frac{\phi_0}{2\pi}\right) \dot{\gamma}^2 - I_c \cos(\gamma) - I\gamma\right]$$  \hspace{0.2cm} (3.2.8)\

We can simplify the Hamiltonian by defining $T$ and $V$ as 

$$T = \frac{1}{2} C_j \left(\frac{\phi_0}{2\pi}\right)^2 \dot{\gamma}^2$$  \hspace{0.2cm} (3.2.9)\

$$V = -\left(\frac{\phi_0}{2\pi}\right)[I_c \cos(\gamma) + I\gamma]$$  \hspace{0.2cm} (3.2.10)\

$$L = T - V$$  \hspace{0.2cm} (3.2.11)\

$$H = T + V$$  \hspace{0.2cm} (3.2.12)\

Where $V$ is defined as the **washboard potential**. Discussed in section 2.7.
### 3.3 Short time expansion

The propagator $K(\phi_Q, t_Q : \phi_A, t_A)$ for $t_Q - t_A > 0$ which connects the flux state $\phi_Q$ at time $t_Q$ and flux state $\phi_A$ at time $t_A$, by Feynman’s path integral formulation [18][19].

$$K(\phi_Q, t_Q : \phi_A, t_A) = \lim_{N \to \infty} \left[ \frac{\mathcal{C}}{2\pi i \hbar \tau} \right]^{N/2} \prod_{j=1}^{N} d\phi(t_j)$$

(3.3.2)

Where $D[\phi(t)]$ is Feynman path differential measure in flux space given by

$$D[\phi(t)] = \lim_{N \to \infty} \left[ \frac{\mathcal{C}}{2\pi i \hbar \tau} \right]^{N/2} \prod_{j=1}^{N} d\phi(t_j)$$

(3.3.2)

and $S_{cl}$ is the classical action given by

$$S_{cl} [\phi(t)] = \int_{t_A}^{t_Q} \left[ \frac{1}{2} \mathcal{C} \left( \frac{d\phi}{dt} \right)^2 - V(\phi) \right] dt$$

(3.3.3)

To observe the classical action in classical field theory on a Euclidean spacetime. Let $\tau = -it$ Therefore, the propagator is modified as

$$K(\phi_Q, \tau_Q : \phi_A, \tau_A) = \lim_{N \to \infty} \left[ \frac{\mathcal{C}}{2\pi i \hbar \tau} \right]^{N/2} \prod_{j=1}^{N} d\phi(t_j)$$

(3.3.4)

The action of particle is given by

$$S_{cl} [\phi(\tau)] = \int_{\tau_A}^{\tau_Q} \left[ \frac{1}{2} \mathcal{C} \left( \frac{d\phi}{d\tau} \right)^2 - V(\phi) \right] d\tau$$

(3.3.5)
CHAPTER 4

PARALLELIZATION AND GPU PROGRAMMING

The purpose of this chapter is to introduce some of the concepts that make GPU programming different from CPU programming. This section will cover the background for parallel computing. Also, some basic terminology will be introduced such as work group, work unit, thread, global size and folding.

4.1 Background

What principally separates Graphical Processing Unit’s (GPU) from Central Processing unit (CPU) is the number of cores. The number of cores decide how many calculations can be carried out simultaneously. GPU contains several Multi Processors (MP), each having an array of cores. From an Architectural point, CPU is composed of just a few cores with lots of cache memory that can deal with a couple of programming strings simultaneously. In contrast, GPU is made from hundreds of cores that can handle thousands of threads at the same time. The capacity of the GPU with 100+ processor cores to processes huge number of threads can accelerate some software by 100x over a CPU. GPU-accelerated computing has now grown mainstream, supported by the latest operating systems from Apple (with OpenCL) and Microsoft (using Direct-Compute). But for the purpose of this thesis we use CUDA [20].
Figure 4.1.1 GPU devotes more transistors for data processing as compared to CPU which has less data processing power. [21].

4.2 Multiprocessors, work groups, Threads and Global sizes.

The number of multiprocessors may vary between different companies and architectures, but all have common arrays of smaller single processors, also known as cores, work units and threads. All the multiprocessors comprise of thread grid.

4.3 Memory

CUDA threads can get data from various memory spaces during execution as shown in figure 4.2. Each thread is assigned a privet local memory. Each Thread Block has shared memory accessible to all threads of the same block and has the same life time as a block. All threads have access to same Global Memory. There are basically four types of memories available on device. Which are global, constant, local and privet.

Global memory is the entire GPU memory. It is also the slowest memory to work with. The global memory is extremely helpful if the kernel code is intended to be re-run, helping the user to transfer data forward and backward between the host and device for each run.
Constant memory is a read only memory and resides in device memory. It is always used with variables that are read only.

Local memory is restricted to each work group. It is frequently used for recurring data as it is faster to access than global memory. The size of this memory however is small, around 16 kB.

Privet memory is like a register, specifically for single threads. It is the fastest memory and smallest in size. Following image shows the memory hierarchy in a GPU system. Image taken from[21].

**Figure 4.3.1** Memory hierarchy in a GPU [21].
4.4 Speedup in parallel programming.

Speedup can be defined as the ratio of the serial runtime of the optimal sequential algorithm for solving a problem to the time taken by the parallel algorithm to solve the same problem on P core processors. For example, by Amdahl’s Law, speedup of a program using multiple processors in parallel is limited by the time needed to execution of serial fraction of the problem. If a problem of size $X$ has serial component $X_s$, the speedup of the problem is

$$S = \frac{X}{(X - X_s) / p + X_s} + X_s$$

As $p \to \infty$

$$S = \frac{X}{X_s}$$

If $X_s = 20\%$, $X - X_s = 80\%$, then

$$S = \frac{1}{0.8 / p + 0.2}$$

$$S \leq \frac{1}{0.2} = 5$$

Therefore, no matter how many processors are used the speedup will not exceed 5.

4.5 Challenges of GPU Implementation

GPU hardware has two noteworthy qualities: raw computing (FLOP’s) and memory bandwidth. Most computational problems can be categorized as one of these two classes.
Grid based algorithms, Monte Carlo, Fast Fourier Transform fall under on the compute/memory bandwidth spectrum depending on system size.

Problems which are too small or unpredictable don’t map well. Small problems do not have the parallelism expected to utilize all the threads on the GPU. Unpredictable problems have an excessive number of significant branches, which can keep information from effectively gushing from GPU memory to the cores or reduce parallelism.

While GPU execution of the algorithms and hypothetical models, various difficulties have emerged and obstructed the way to progress. Most important are the run-time and memory restrictions.

4.6 Kernel runtime limitation:

Graphics Processing Units are primarily designed for computing data for visual and graphical objects on screen. This aspect implies that there should be a run time limitation to prevent the graphics card from calculating too long before giving the output. This feature is particularly important and necessary to prevent overheating of the system which intern may damage the hardware.

Common limitation on Nvidia GPU systems is of 5 seconds run time per kernel. Therefore, the kernel code should be executed within 5 seconds. While executing scientific problems which may take hours to compute, the 5 second run time is not sufficient. We use some alternatives to overcome this issue.

1. Add an extra GPU only for execution of non-graphical computations.

2. Split your kernel into several small kernels. This is a widely recommended solution as adding a new GPU is costly.
CHAPTER 5

EUROPEAN OPTION PRICING FOR BLACK-SHOLES MODEL

5.1 Theory

In computational finance, the Black–Scholes equation is a partial differential equation (PDE) central to the price evolution of a European call or European put under the Black–Scholes model. Key is related understanding of financial insight behind the equation is that one can hedge the option by buying and selling the original asset in just the right way and consequently “eliminate risk”. This implies that there is single right price for the option, as given back by Black–Scholes formula. Black-Sholes equation is widely used with some adjustments and corrections, by options market participants.

Option trading is spreading worldly since 1973 when CBOE (Chicago Board of Options Exchange) was formally established by American Government. As the trading volume increases, option brokers deal with a new rising problem, option pricing. Researches then carried out to and the most effective way to determine option price. Model to predict option price, Black-Scholes Model was invented [23].

While using Black-Scholes model we must make some assumptions:

1. It is a European option and can only be worked out at time of expiration.
2. No profits are paid out throughout the life of the option.
3. The risk-free rate and volatility of the option are constant.

This chapter covers European vanilla option pricing derivation of Black-Sholes formula. Consider the price of a European Call option \( C(S,t) \), \( S \) spot price, \( K \) strike price, \( r \) risk free interest rate, \( \sigma \) volatility of thee asset and \( T \) is the time for maturity. \( N \) is the
Cumulative Distribution Function (CDF) of the standard distribution. The closed form solution of the Black-Sholes equation for $C(S, t)$ is shown below [15].

$$C(S, t) = SN(d_1) - Ke^{-rT}N(d_2)$$ \hspace{1cm} (5.1.1)

Where $d_1$ and $d_2$ are defined as:

$$d_1 = \frac{\log\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma \sqrt{T}}$$ \hspace{1cm} (5.1.2)

$$d_2 = d_1 - \sigma \sqrt{T}$$ \hspace{1cm} (5.1.3)

Similarly, for put-call parity, we can price a European vanilla put option $P(S, t)$ is given by:

$$P(S, t) = Ke^{-rT} - S + C(S, t)$$ \hspace{1cm} (5.1.4)

$$P(S, t) = Ke^{-rT} - S + (SN(d_1) - Ke^{-rT}N(d_2))$$ \hspace{1cm} (5.1.5)

Now we realize the function $N$, The cumulative distribution function of the standard normal distribution. Formula is given by:

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$$ \hspace{1cm} (5.1.6)

Notice that $N$ is closed form solution. In order to calculate the closed form we need the probability density function of the standard distribution function which can be given as:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$ \hspace{1cm} (5.1.7)
5.2 Path integral model of European option pricing for Black-Scholes equation

A new way to approach Black-Scholes model and has a huge potential to modify it is Path Integral. Richard P. Feynman published the path integral method to connect classical physics and quantum mechanics. Remarkably, this method could be applied not only in quantum mechanics, but also in finance sector[23].

One basic equation which becomes an analogy between the quantum and the mechanics is Newton’s Second Law which corresponds to the equation Schrodinger:

\[-\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} + U(x)\psi = i\hbar \frac{\partial \psi}{\partial t}\]  
(5.2.1)

Classical action S expressed with integral of Lagrange function, \(L = K - U\) with \(K\) as kinetic energy and \(U\) as potential energy, with the respect of time from initial to final of the time-frame.

\[S = \int_{t_i}^{t_f} L dt \approx L_{avg} \varepsilon, \varepsilon = \Delta t\]  
(5.2.2)

\(L_{avg}\) is the Lagrange with the kinetic and potential energy in their simplest case represented in approximation.

\[K = \frac{1}{2} m \left( \frac{x_{f} - x_{i}}{\epsilon} \right)^2\]  
(5.2.3)

\[U = U\left( \frac{1}{2} (x_{f} + x_{i}) \right)\]  
(5.2.4)

After doing the integral function we get Schrodinger Equation as described in Equation above.
We use Feynman's path integral process to derive Black-Scholes equation. Black-Scholes equation is a stock option model to done as option price in certain time from its initial condition. Mathematically, we can form this transformation

\[ C(S_f, t_f) = \int_{-\infty}^{\infty} G(S_i, S_f)C(S_i, t_i)dS \]  \hspace{1cm} (5.2.3)

Where \( G(S_i, S_f) \) is the kernel or the propagator function. \( S \) is the stock price, \( C(S_i, t_i) \) in economics signify option's price is an analogy to the wavefunction of the particle in certain position in time.

Black-Scholes Model with its discussion equation have similarities with Schrodinger equation form in Quantum Mechanics.

Suppose we get the qubits \( |\psi> \) solving the partition function. It will be a derivative free function. What if we do it in a parallel programming environment where the cross talk between the threads with the qubits being entangled and executed in GPU environment.

This can be used for building quantum gates or solving real world problems such as implementation of Black-Sholes equation for quantum finance.
CHAPTER 6

RESULTS

6.1 Implementation of Black-Sholes in CUDA:

The program consists the kernel code that contains actual calculations that are to be performed and runs on a GPU. This will give a basic understanding of CUDA functions used to allocate memory and transfer data back and forth the graphics card.

Once we have the right libraries imported, we begin to make statistical functions which make up the vast majority of the calculations for prices. Standard normal probability density function is realized. \( f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \)

Next statistical function is cumulative distribution function. The code is an approximation function. It also is a recursive function which call itself.

Final set of functions are associated with pricing of European vanilla calls and puts. We calculate d1 and d2. we can calculate the closed-form solution for the European vanilla call price, similarly for vanilla put price.

Then we allocate N elements each with N number of threads. Here we set N to 3. And allocate vectors in device memory using cudaMalloc function. Then copy input from host memory to device memory using cudaMemcpy function, pass the appropriate number of blocks and threads. Call the kernel to execute, copy result from device to host. Finally, we output the parameters and prices.
Figure 6.1.1 Output of the CUDA code in command prompt.

6.1.1 Code:

```cpp
#define _USE_MATH_DEFINES

#include <iostream>
#include <cmath>

// Standard normal probability density function
__device__ double norm_pdf(const double x) {
    return (1.0/(pow(2*M_PI,0.5)))*exp(-0.5*x*x);
}

// An approximation to the cumulative distribution function
// for the standard normal distribution
// Note: This is a recursive function
__device__ double norm_cdf(double x) {
    int neg_flag = 0;
    if (x < 0) {
```
\[
x = -x;
\]
\[
\text{neg\_flag} = 1;
\]
\[
\text{double } k = 1.0/(1.0 + 0.2316419*x);
\]
\[
\text{double } k\_sum = k*(0.319381530 + k*(-0.356563782 + k*(1.781477937 + k*(-1.821255978 + 1.330274429*k))));
\]
\[
\text{double } result;
\]
\[
\text{result} = (1.0 - (1.0/(\text{pow}(2*\text{M}_\Pi,0.5)))*\text{exp}(-0.5*x*x) * k\_sum);
\]
\[
\text{if } (\text{neg\_flag} == 0) \{ \\
\quad \text{return } \text{result}; \\
\} \text{ else } \{ \\
\quad \text{return } 1.0 - \text{result}; \\
\}
\]

// This calculates d_j, for j in {1,2}. This term appears in the closed
// form solution for the European call or put price

__device__ double d_j
(const int j, const double S, const double K, const double r, const double v, 
const double T) 
{ 
\quad \text{return } (\log(S/K) + (r + (\text{pow}(-1,j-1))*0.5*v*v)*T)/(v*(\text{pow}(T,0.5))); 
}

// Calculate the European vanilla call price based on
// underlying S, strike K, risk-free rate r, volatility of
// underlying sigma and time to maturity T
__global__ void call_price
  (double *call, const double *S, const double *K, const double *r, const double *v,
   const double *T, int N) {

    int i = blockDim.x * blockIdx.x + threadIdx.x;
    if(i < N)
      call[i] = S[i] * norm_cdf(d_j(1, S[i], K[i], r[i], v[i], T[i])) -
                 K[i]*exp(-r[i]*T[i])
                 * norm_cdf(d_j(2, S[i], K[i], r[i], v[i],
                               T[i]));
  }

// Calculate the European vanilla put price based on
// underlying S, strike K, risk-free rate r, volatility of
// underlying sigma and time to maturity T
__global__ void put_price
  (double *put, const double *S, const double *K, const double *r, const double *v,
   const double *T, int N) {

    int i = blockDim.x * blockIdx.x + threadIdx.x;
    if(i < N)
      put[i] = -S[i] * norm_cdf(-d_j(1, S[i], K[i], r[i], v[i],
                                  T[i]))+K[i]*exp(-r[i]*T[i])
                 * norm_cdf(-d_j(2, S[i], K[i], r[i], v[i],
                                  T[i]));
  }
/Array of N elements each
//number of threads N
//this code sets N=3
int main(int argc, char **argv) {

    // First we create the parameter list
    int N = 3;
    int size = N * sizeof(double);
    double *S = (double *) malloc(size);
    double *K = (double *) malloc(size);
    double *r = (double *) malloc(size);
    double *v = (double *) malloc(size);
    double *T = (double *) malloc(size);
    double *call = (double *) malloc(size);
    double *put = (double *) malloc(size);

    //initialize dummy input (should get data from a database to be useful)
    for(int i=0; i < N; i++){
        S[i] = 100.0+i; // Option price
        K[i] = 100.0+i; // Strike price
        r[i] = 0.05+i; // Risk-free rate (5%)
        v[i] = 0.2+i; // Volatility of the underlying (20%)
        T[i] = 1.0+i; // One year until expiry
    }

    //Allocate vectors in device memory
    double *d_S;
    cudaMalloc(&d_S,size);
    double *d_K;
    cudaMalloc(&d_K,size);
double *d_r;
cudaMalloc(&d_r,size);

double *d_v;
cudaMalloc(&d_v,size);

double *d_T;
cudaMalloc(&d_T,size);

double *d_call;
cudaMalloc(&d_call,size);

double *d_put;
cudaMalloc(&d_put,size);

//copy input from host memory to device memory
cudamempcy(d_S,S,size,cudamempcyHostToDevice);
cudamempcy(d_K,K,size,cudamempcyHostToDevice);
cudamempcy(d_r,r,size,cudamempcyHostToDevice);
cudamempcy(d_v,v,size,cudamempcyHostToDevice);
cudamempcy(d_T,T,size,cudamempcyHostToDevice);

//pass the appropriate num of blocks and threads
int threadsPerBlock = 256;
int blocksPerGrid = (N + threadsPerBlock - 1) / threadsPerBlock;

call_price<<<blocksPerGrid,threadsPerBlock>>>(d_call,d_S,d_K,d_r,d_v,d_T,N);

put_price<<<blocksPerGrid,threadsPerBlock>>>(d_put,d_S,d_K,d_r,d_v,d_T,N);

//copy result from device to host
cudaMemcpy(call,d_call,size,cudamempcyDeviceToHost);
cudaMemcpy(put,d_put,size,cudamempcyDeviceToHost);
// Finally we output the parameters and prices

for(int i = 0; i<N; i++){
    std::cout << "Underlying:      " << S[i] << std::endl;
    std::cout << "Strike:          " << K[i] << std::endl;
    std::cout << "Risk-Free Rate:  " << r[i] << std::endl;
    std::cout << "Volatility:      " << v[i] << std::endl;
    std::cout << "Maturity:        " << T[i] << std::endl;
    std::cout << "Call Price:      " << call[i] << std::endl;
    std::cout << "Put Price:       " << put[i] << std::endl;
}

return 0;

6.2 Implementation of RSCJ Josephson Junction in MATLAB:

A single Josephson junction is a closed loop circuit with an external current flowing thought it. The Josephson junction was modelled. Which was solved for different values of $\alpha$ and $\eta$. In the output, we have different phases as the current traveling through the JJ increases from 0A to 2 times the critical current. We can see that when the current introduced is set to 0, it produces no phase difference. That is to be expected as 0 current will act as a DC source traveling through the junction. And as current is increased, the phase becomes more pronounced and constantly varying, meaning that the phase has
shifted forward leading the original signal. This is the representation of the tilted washboard potential explained in section 2.3. The equation was solved using MATLAB solver ode23s. The equation was first reformed into a first-order differential equation with two unknown variables. A current-biased Josephson Junction employs or creates a “washboard” shaped potential. Splitting in the wells indicates allows us to use the lowest two levels as qubit states. The higher energy state |1> can be detected because the tunneling probability under a microwave probe will be 500 times as probable to induce a transition.

Figure 6.2.1 Phase verse time graph with Washboard potential with zero bias and non-zero bias.
Figure 6.2.2 Zoomed Phase verse time graph with Washboard potential with zero bias and non-zero bias

6.2.2 Code:

```matlab
tspan = 0:0.01:100;
Ic = 1e-7;
A = 1/100;
counter = 0;
for I = 0:Ic/4:2*Ic
    [t,x] = ode23s(@(t,x)method5(t,x,A,I,Ic),tspan,[1 0]);
    plot(t,deg2rad(x(:,2)),'DisplayName',sprintf('%1.2d',I)),grid on,hold on;
    title('Phase vs. Time');xlabel('Time(s)');ylabel('Phase(rad)');counter=counter+1;
end
legend('show');```

45
Here, there are three Josephson junctions set in series. There is no phase difference from the initial Josephson junction given the current is set at 0.

![Graph of phase verse time for three Josephson junctions in series.](image)

**Figure 6.2.3** Graph of phase verse time for three Josephson junctions in series.

### 6.2.3 Code:

```matlab
% Figure 6.2.3 Code:

tspan = 0:0.01:100;
Ic1 = 1e-7;
Ic2 = 1e-5;
Ic3 = 2e-4;
I = 0;
A = 1/100;

counter = 0;
```
\[ t, x \] = \text{ode23s}\left( @(t,x)\text{method6}(t,x,A,I,Ic1),tspan,[1 \ 0]\right); \\
\[ t, y \] = \text{ode23s}\left( @(t,x)\text{method6}(t,x,A,I,Ic2),tspan,[1 \ 0]\right); \\
\[ t, r \] = \text{ode23s}\left( @(t,x)\text{method6}(t,x,A,I,Ic3),tspan,[1 \ 0]\right); \\

\text{plot}(t,\text{deg2rad}(x(:,2)),'\text{DisplayName}',\text{sprintf('1d',I)),'grid on,hold on;} \\
\text{plot}(t,\text{deg2rad}(y(:,2)),'\text{DisplayName}',\text{sprintf('1d',I));} \\
z = r(:,2)-x(:,2); \\
\text{plot}(t,z,'\text{DisplayName}','\text{Phase r-x}'); \\
\text{title('Phase vs. Time');xlabel('TIme(s)');ylabel('Phase(rad)');} \\
\text{legend('show');}
CHAPTER 7

CONCLUSIONS

We have presented three techniques to generate qubits for the quantum virtual machine namely phase, flux, and charge qubit with the Hamiltonian model. We analyzed the equivalent circuits of each of them in great details and implemented the phase qubit model of one Josephson junction to give the washboard potential. We analytically rewrote the second order differential equations to first order differential equations. These were solved using Matlab. Energy levels are found in the well. The first two levels can be used for qubit states. The qubit circuit is based on the possibility that states in the potential can tunnel through the potential.

The Josephson junctions are key ingredients in any superconducting qubit design using a Hamiltonian approach. However, it also results in decoherence and noise. The Feynman approach of path integral model eliminates these decoherence. We do not implement this approach in this thesis but do a mathematical analysis.

Black-Scholes Model has similarities with Schrodinger equation form in Quantum Mechanics. We use parallel programming CUDA to solve the Black-Sholes European option pricing equation in classical form to show that parallel programming can be used for faster computations to solve mathematical models.
CHAPTER 8

FUTURE WORK

The next step for the Josephson junction is to implement the path integral model which in turn gives qubits that it can be coupled to form a quantum gate in a parallel environment of CUDA where the cross talk between the threads with the qubits being entangled and executed in GPU environment. We only use European vanilla option solution for Black-Sholes equation. Next step would be to implement this equation in path integral model in the GPU environment for faster computations. By doing this we hope to link the qubits generated from the path integral model of the Josephson junction to the Black-Sholes path integral model in a parallel programming environment.


[14] NanoSQUIDs: Basics & recent advances Maria José Martínez-Pérez Published Online: 2017-08-02 | DOI: https://doi.org/10.1515/psr-2017-5001

[15] Phase-locked spiking of inner ear hair cells and the driven noisy Adler equation Roie Shlomovitz, Yuttana Roongthumskul, Seung Ji, Dolores Bozovic, Robijn Bruinsma


[19] Semiclassical treatment of the path-integral approach to special effects in superconductivity


