Extending and Simulating the Quantum Binomial Options Pricing Model

A thesis presented
by

Keith Meyer

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of

Computer Science

The University of Manitoba
Winnipeg, Manitoba
January 2009

Thesis advisor
Author

Dr. William Kocay
Keith Meyer

Extending and Simulating the Quantum Binomial Options Pricing Model

Abstract

Pricing options quickly and accurately is a well known problem in finance. Quantum computing is being researched with the hope that quantum computers will be able to price options more efficiently than classical computers. This research extends the quantum binomial option pricing model proposed by Zeqian Chen to European put options and to Barrier options and develops a quantum algorithm to price them. This research produced three key results. First, when Maxwell-Boltzmann statistics are assumed, the quantum binomial model option prices are equivalent to the classical binomial model. Second, options can be priced efficiently on a quantum computer after the circuit has been built. The time complexity is $O((N-\tau)\log_2(N-\tau))$ and it is in the **BQP** quantum computational complexity class. Finally, challenges extending the quantum binomial model to American, Asian and Bermudan options exist as the quantum binomial model does not take early exercise into account.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to begin by thanking my committee, my advisor Dr. Kocay, my computational finance professor Dr. Tulsi, Dr. Southern for his help with quantum statistics, Dr. Zeqian Chen for his comments on my early work on the subject, Dr. Peter Shor for his early direction, Mike van de Vijsel, Doug Holmes, Amanda Krasey and my other colleagues for their feedback, my parents Walter and Lorna, my brother Mike, my wife Alana, my children James and Alora and all the people who have supported me along the way.

This thesis is dedicated to those who pursue reality.

Chapter 1

Introduction

Many of the problems facing the finance community have no known analytical solution. As a result numerical methods and computer simulations for solving these problems have proliferated. This research area is known as *computational finance.* Many computational finance problems have a high degree of computational complexity and are slow to converge to a solution on classical computers. In particular, when it comes to option pricing, there is additional complexity resulting from the need to respond to quickly changing markets. For example, in order to take advantage of inaccurately priced stock options, the computation must complete before the next change in the almost continuously changing stock market. As a result, the finance community is always looking for ways to overcome the resulting performance issues that arise when pricing options. This has lead to research that applies alternative computing techniques to finance. One of these alternatives is *quantum computing.* Just as physics models have evolved from classical to quantum, so has computing. Quantum computers have been shown to outperform classical computers when it comes to simulating quantum mechanics [8] as well as for several other algorithms such as Shor's factorization algorithm [32] and Grover's quantum search algorithm [16], making them an attractive area to research for solving computational finance problems. Following these breakthroughs, Chen published a paper in 2001 [10] where he presents a *quantum binomial options pricing model* or simply abbreviated as the *quantum binomial model*. There were several key results in his paper. First of all, he shows a quantization of the classical Black-Scholes-Merton [7] based binomial option pricing model developed by Cox-Ross-Rubinstein [12] for European options. He then shows his quantum binomial model does not pose the risk indifference paradox that appears in the classical binomial model. Next he shows its risk-neutral world exhibits a structure as a disk in the unit ball of $\mathbf{R}^3$ with a radius that is a function of the risk-free interest rate with two thresholds that prevent arbitrage (risk-less profit) opportunities. Finally, he suggests that the quantum binomial model may be implemented using quantum computers.

The first part of this research extends Chen's quantum binomial model from European Call options to various styles of vanilla and exotic options by deriving the respective quantum mechanical binomial equations. Each of the equations is then analyzed and evaluated. The second part of this research is the implementation of several of the option pricing formulas on a quantum circuit simulator. This is done in order to verify not only the accuracy and performance of calculating the prices, but all of the claims Chen made regarding its structure. The final part of this research compares Chen's quantum binomial model to the classical Cox-Ross-Rubinstein model.

The rest of this thesis is setup as follows: First of all, a brief overview of quantum computing and finance theory is presented. This is followed by a related work section that provides a brief overview of relevant quantum finance research to date and a detailed look at Chen's quantum binomial model. This is then followed by a detailed presentation of the results of this research, and finally, a conclusion.

1.1 Background on Quantum Computing

1.1.1 Computational Complexity

In 1982, Richard Feynman suggested that modeling computers on the principles of quantum mechanics would intrinsically allow us to overcome the difficulties classical computers have simulating quantum mechanical systems [15]. This is now known as quantum computing. When compared to a classical computer, it has been shown that exponential speed up can be achieved by simulating quantum mechanics on a quantum computer [8]. Thus, if finance is quantum mechanical, perhaps it too, can be simulated with exponential speed on a quantum computer. Spurred by these results, researchers looked to find ways to speed up algorithms not related to simulating quantum mechanics. Eventually this lead to several significant quantum algorithm discoveries in the mid-1990s. Most notably are Shor's factorization algorithm in 1994 [32] and Grover's quantum search algorithm in 1996 [16]. Shor's algorithm, shown in Figure 1.1, can factor a number N in $O((\log N)^3)$ time using $O(\log N)$ space, where no classical algorithm is known that can factor a number N in time $O((\log N)^k)$ for any k. Grover's algorithm can be used to search an unsorted database with N entries in $O(\sqrt{N})$ time using $O(\log N)$ storage space, where the

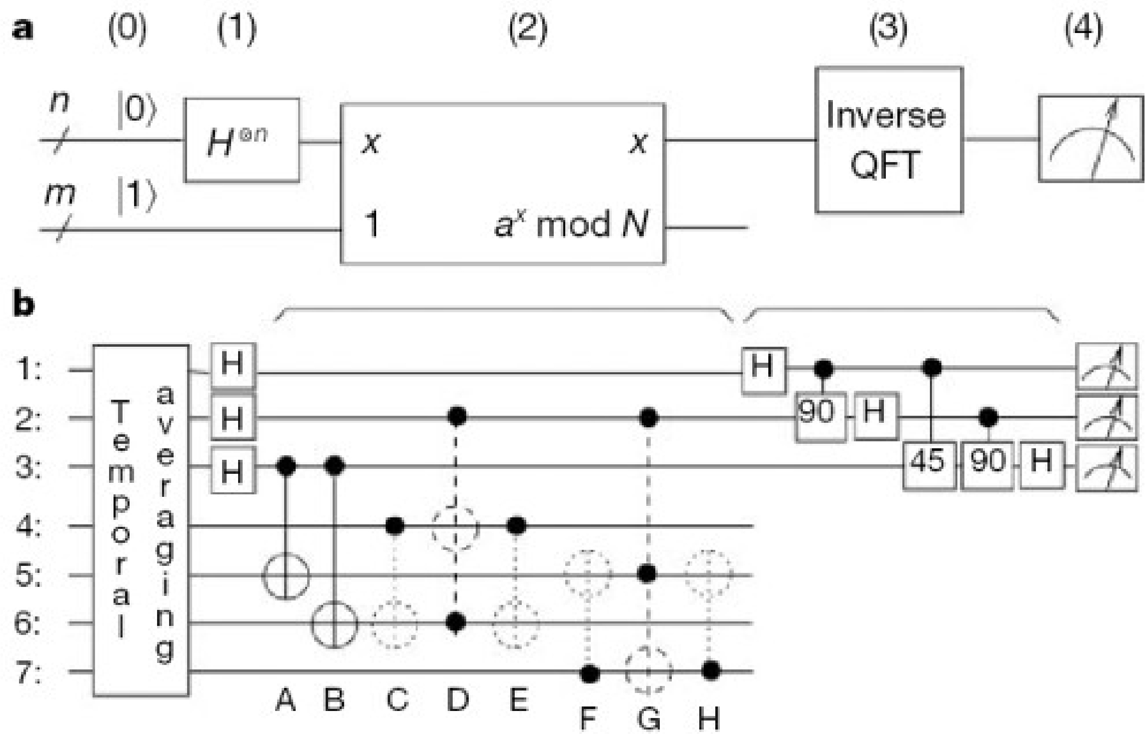


Figure 1.1: Quantum Circuit implementing Shor's Algorithm. Picture by The Center for Bits and Atoms [1].

best classical algorithm uses $O(N)$ storage. The efficiency of these quantum algorithms and others [22] have spawned a great deal of interest in quantum computers. That said, even though there are quantum algorithms that have been developed that perform better than any known classical algorithm, there is still no proof that an equivalent classical algorithm does not exist. Further, there has been no evidence that quantum computers have algorithms that would change the complexity classification of any problems [29]. Such a discovery would be very significant and would help move towards answering one of largest open questions in computer science — does $\mathbf{P} = \mathbf{NP}$? Note, however, that there are still very few quantum algorithms and that developing new ones that are more efficient than their classical counterparts is proving to be quite difficult. There are also quantum algorithms that are less efficient than their classical counterparts but are still useful because they run on a quantum computer and can be used as a subcomponent of a larger quantum algorithm that is more efficient overall. Even with these questions outstanding and challenges, the results look promising and research continues.

Not surprisingly, quantum complexity theory has been researched with the arrival of quantum computing. There has been a fair amount of research into quantum complexity theory, but for purposes of this research we will focus on a few of the essential results. The class of problems that can be efficiently solved by a quantum computer is called **BQP**, for "bounded error, quantum, polynomial time". Essentially, it stands for the class of decision problems that can be solved with a bounded probability of error, using a polynomial-size quantum circuit. Another way of looking at it is, a quantum computer is said to efficiently solve a problem if its answer will be right with high probability for every instance. If that solution runs in polynomial time, then that problem is in **BQP**. On the other hand, **BPP** for "Bounded-error, Probabilistic, Polynomial time" is the class of decision problems solvable in polynomial time, with an error probability of at most 1/3 for all instances on a classical computer. Considering quantum computers only run probabilistic algorithms, **BQP** for a quantum computer is the equivalent of **BPP** for a classical computer. Further, **BQP** is contained in the complexity class **P**, which is a subclass of **PSPACE** where **PSPACE** is the class of problems that can be solved by algorithms which only need a polynomial amount of memory to run. Finally, **BQP** is suspected not to be part of **NP**-complete and to be a strict superset of **P**, however this has not been proven. To summarize:

$$\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{BQP} \subseteq \mathbf{PSPACE} \tag{1.1}$$

For a more comprehensive examination of quantum complexity theory see Watrous [36].

1.1.2 Quantum Computers and Quantum Circuits

There are several differences between classical and quantum computers. Classical computers are based on bits that equal either 0 or 1. On the other hand, Quantum computers are based on what are called *qubits*. Qubits differ significantly from classical bits as they are in a probabilistic state in the two-dimensional complex vector space and written as follows:

$$\alpha |0\rangle + \beta |1\rangle \tag{1.2}$$

and can be envisioned as a Bloch Sphere [17]. In quantum mechanics, the state of a physical system is identified by a point in the Hilbert space $\mathcal{H}$ of the system. Each vector in the Hilbert space is called a ket and written $|\psi\rangle$. The ket can be viewed as a column vector and written out as $|\psi\rangle = (c_0, c_1, c_2, ...)^T$ (where T means transpose) for a given basis when, as is in quantum computing, the Hilbert space is finite-dimensional. Every ket $|\psi\rangle$ has a dual vector bra written as $\langle\psi|$, that can be expressed as the row vector $\langle\psi| = (c_0^*, c_1^*, c_2^*, ...)$ where c^* means the complex conjugate of c. When the qubit is measured, $|\alpha|^2$ is the probability that the qubit will be equal to 0 and $|\beta|^2$ is the probability that the qubit will be equal to 1, and by the law of total probability, $|\alpha|^2 + |\beta|^2 = 1$. For example, a qubit is often initialized to the following state where it has a 50% chance of being either a 0 or 1 when measured:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{1.3}$$

Multi-qubit systems also have probabilistic states as shown with these 2 qubits:

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \tag{1.4}$$

An example of a two qubit state is as follows:

$$\frac{1}{\sqrt{4}}|00\rangle + \frac{1}{\sqrt{4}}|01\rangle + \frac{1}{\sqrt{5}}|10\rangle + \frac{\sqrt{3}}{\sqrt{10}}|11\rangle \tag{1.5}$$

which means that there is a 25% chance the state will be $|00\rangle$, a 25% chance the state will be $|01\rangle$, a 20% chance the state will be $|10\rangle$ and a 30% chance the state will be $|11\rangle$ when the qubits are measured. Another key concept in quantum computing is the tensor product[1], which is indicated by $\otimes$. The main purpose of the tensor product in physics, is, to represent composite quantum mechanical systems. In other words, it is used to represent the combination of multiple quantum mechanical systems. This is illustrated with equation 1.6 that shows the tensor product of two systems where each system is represented by the outer product of two basis vectors and alternatively by equation 1.7 that shows the tensor product of two systems where each system is

[1] Also known as the Kronecker product.

represented by a 2×2 matrix.

$$\begin{aligned}
|u\rangle \langle u| \otimes |v\rangle \langle v| &= \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} \times \begin{bmatrix} u_{11}^* & u_{21}^* \end{bmatrix} \otimes \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} \times \begin{bmatrix} v_{11}^* & v_{21}^* \end{bmatrix} \\
&= \begin{bmatrix} u_{11}u_{11}^* & u_{11}u_{21}^* \\ u_{21}u_{11}^* & u_{21}u_{21}^* \end{bmatrix} \otimes \begin{bmatrix} v_{11}v_{11}^* & v_{11}v_{21}^* \\ v_{21}v_{11}^* & v_{21}v_{21}^* \end{bmatrix} \qquad (1.6)
\end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \otimes \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} &= \begin{bmatrix} u_{11} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} & u_{12} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \\ u_{21} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} & u_{22} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \end{bmatrix} \qquad (1.7) \\
&= \begin{bmatrix} u_{11}v_{11} & u_{11}v_{12} & u_{12}v_{11} & u_{12}v_{12} \\ u_{11}v_{21} & u_{11}v_{22} & u_{12}v_{21} & u_{12}v_{22} \\ u_{21}v_{11} & u_{21}v_{12} & u_{22}v_{11} & u_{22}v_{12} \\ u_{21}v_{21} & u_{21}v_{22} & u_{22}v_{21} & u_{22}v_{22} \end{bmatrix}
\end{aligned}$$

Qubits are used to store information during the execution of a Quantum algorithm. Quantum algorithms themselves are often described using what are known as quantum circuits. Quantum circuits are a way of representing quantum algorithms just like classical digital circuit diagrams represent classical algorithms. Quantum circuits are made up of a variety of gates but instead of the familiar classical AND and OR gates, there are quantum gates called CNOT, Hadamard, rotation and more.

Each of these quantum gates can be represented by a unitary matrix[2] and are the building blocks of the quantum circuit. As shown in Figure 1.2, and unlike classical circuits, quantum circuits execute only by moving from left to right. During each step of the circuit's execution, it builds an effective matrix by moving top to bottom by performing cumulative matrix-matrix tensor products. Using the effective matrix, the execution then moves across the circuit from left to right performing vector-matrix multiplication. These types of linear algebra operations are well known to be difficult for classical computers and provide insight into why quantum computers, which do these operations intrinsically, have been shown to solve some quantum mechanical problems faster as discussed earlier. Further, as discussed above, if finance can be shown to be quantum mechanical, then simulating it on a quantum computer should also be intrinsically faster than simulating it on a classical computer. During the execution of a quantum algorithm, the quantum state must not become decoherent. Quantum decoherence occurs when a system in a quantum mechanical state starts to interact with an external system and begins to tend towards classical behavior. Once the quantum mechanical system that is supporting the quantum computer begins to act classically, the quantum computation running on the quantum computer typically fails. This is because quantum decoherence causes the probabilistic state of the quantum system to be lost and thus the probabilistic state cannot be leveraged by quantum algorithms. On the contrary, quantum decoherence is required during measurement. Measurement is a necessary step in obtaining the results of a quantum computation, but it is typically performed at the end of a quantum algorithm,

[2] A unitary matrix is a n by n complex matrix U that satisfies the condition $(U^T)^*U = U(U^T)^* = I_n$ where I_n is the identity matrix.

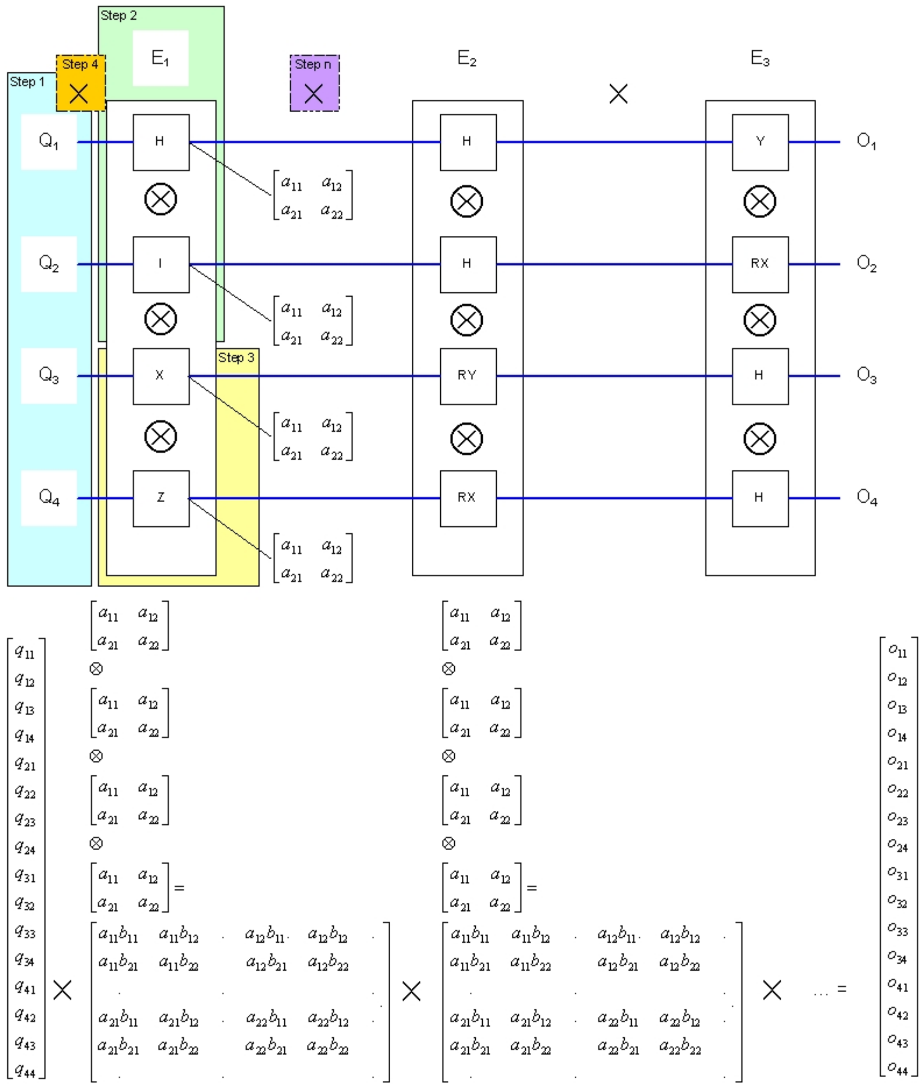


Figure 1.2: Execution of a Quantum Circuit

when all of the computations have been completed. Exceptions to this rule are the hybrid quantum/classical algorithms like teleportation [9] that rely on the measurement of certain qubits during the execution. As will be shown later on, the idea of measurement is central to the quantum binomial model.

As mentioned above, one other key to note is that quantum computing is currently restricted to finite dimensional Hilbert spaces. That means that problems relating to infinite dimensional Hilbert spaces such as those found in quantum field theory are not intrinsically efficiently solved on quantum computers. Further, to date, there is no known way to efficiently simulate quantum field systems on a quantum computer. As will be shown in the results section, path dependent options such as Asian options appear to actually be quantum field problems.

1.1.3 Statistical Mechanics

In the field of statistical mechanics there are both quantum and classical statistics. Statistics consider the distinguishability of particles and how that affects the number of unique states the particles can create. Maxwell-Boltzmann statistics are used to describe the statistics of *distinguishable* classical particles. In other words the configuration of particle P_1 in energy state A and particle P_2 in energy state B is not the same as the configuration of particle P_2 in energy state A and particle P_1 in energy state B. Extending this to N particles yields the Maxwell-Boltzmann distribution of particles in energy states. On the other hand, Bose-Einstein and Fermi-Dirac statistics are used when quantum effects have to be taken into account and the particles are considered *indistinguishable*. When particles are indistinguishable, the number of unique states is decreased considering the number of unique configurations is reduced. Bose-Einstein statistics apply to bosons and Fermi-Dirac statistics apply to fermions and both become Maxwell-Boltzmann statistics at high temperatures or low concentrations. See Figure 1.3 for a detailed picture of the difference between these statistical models. In the quantum pricing model, the particles are the N qubits

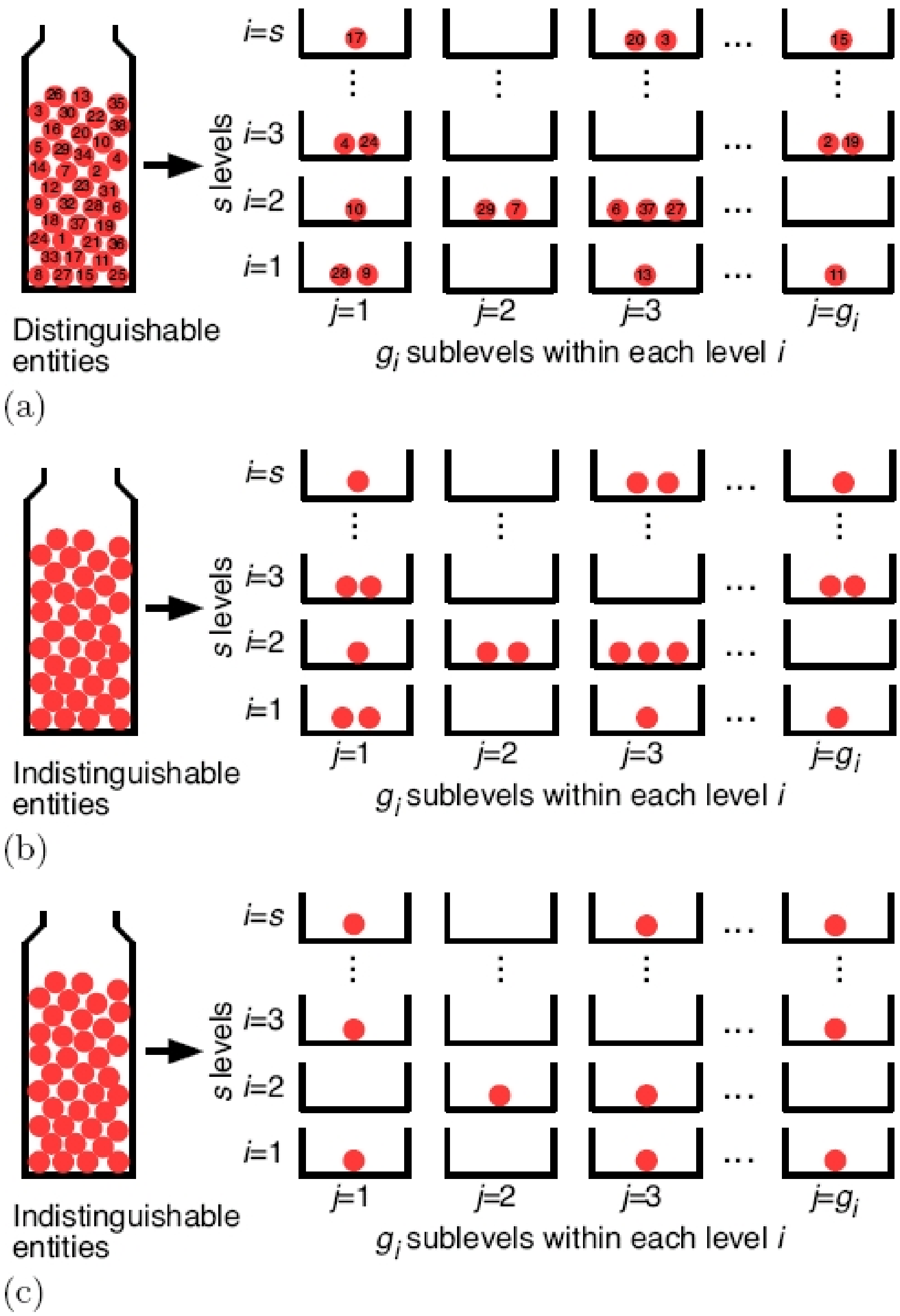


Figure 1.3: Configurations of (a) Maxwell-Boltzmann, (b) Bose-Einstein and (c) Fermi-Dirac ball-in-box models. Note that all N entities are fully allocated. Image taken from Niven and Grendar [30].

used to simulate the N pricing periods on a quantum computer. Further, the model is general enough that Maxwell-Boltzmann, Bose-Einstein or Fermi-Dirac statistics can be used. If classical Maxwell-Boltzmann statistics are used, the quantum binomial model collapses into the classical binomial model. However, if either of the quantum statistics are used, the quantum model will produce an option price that is different from the classical one. Determining the usefulness of this new quantum option price has been left to future research.

1.1.4 Background on Finance

Much of finance today is concerned with markets, risk and reward. A market can be a publicly accessible place where trading takes place, like a stock or futures exchange, or it can be a network that links buyers and sellers, like the over-the-counter market. Markets facilitate investors wishing to manage their risks and by doing so, they also manage their potential rewards and losses. Typically, an investor expects a higher possible reward as they take on more risk. Finding a good balance of risk and reward for a particular investment and investor are essential. Having the ratio too high can mean never receiving the reward and potentially taking large losses, whereas having the ratio too low can mean that the cost of the investment is such that you would have been better off not investing. Trading options is an essential part of managing risk and reward and so is understanding the fundamentals of the financial instruments and processes used by exchanges explained in the sections below.

Securities

Securities are financial instruments that indicate ownership, a debt agreement or the right to own. Some examples of each are shown in Table 1.1 below. Securities

Securities Category	Example
Ownership	Common Stock, Preferred Stock
Debt Agreement	Bond, Note, Certificates, Mortgages
Right-to-own	Option, Future, Swap, Warrant

Table 1.1: Examples of Securities

represent a contract between parties that can be valued and traded. The structure of each security and the types of contracts that each security can be entered into vary and influence the ability to manage the risk/reward ratio. All right-to-own securities are useful tools for managing an investor's risk/reward ratio, but as shown in the following sections, options are considered the most useful.

Exchanges

There are several exchanges in the world trading various securities. Table 1.2 shows some of the top exchanges by market capitalization as of September 2005 [37]. Trading on exchanges is done by either using the open outcry system or electronic trading. In the open outcry system, traders physically meet and use a series of hand gestures and verbal communication to complete trades. On the other hand, electronic trades are completed using computers. This highlights the significance of being able to calculate an accurate option price quickly — if a mispriced option can be found on the market, it can be automatically traded using a computer for a profit.

Exchange	Market Cap (Billions)
NYSE	$12992.1
Tokyo Stock Exchange	$4042.6
Nasdaq	$3475.2
London Stock Exchange	$2988.2
Osaka Stock Exchange	$2632.2
Euronext	$2607.3
TSX Group (Toronto)	$1361.5
Deutsche Borse	$1185.3
BME Spanish Exchanges	$1013.8
Hong Kong Exchange	$981.7
Swiss Exchange	$896.1
Australian Stock Exchange	$777.7
OMX Exchange	$747.3
Borsa Italiana	$730.5
Korea Exchange	$574.9
Bombay Stock Exchange	$512.8
JSE (South Africa)	$475.0
Sao Paulo Stock Exchange	$437.1

Table 1.2: Market Capitalization of Top Exchanges

Options

As mentioned above, options are a crucial tool for managing an investor's risk/reward ratio. Options can be combined with other securities so the investor can customize the risk/reward ratio of their portfolio. The portfolio can even be setup in such a way that the risk is zero (called a perfect hedge). Options are financial instruments that give the owner the right to execute a future transaction on some security such as a stock or futures contract — if they choose to do so. As an example, consider the purchase of what is known as a call option. A call option gives the owner of the option the right to buy a specific number of shares of a security at a set price called the strike price until some set expiry date. On the other hand, the party who sold the option is obligated by the terms of the option to sell the owner of the option the underlying security for the strike price before the expiry date. Call options can be purchased for several reasons. Two of which are, buying a call option to benefit from a rise in the price of the underlying security or to lock in the price of the underlying security. The trader will estimate the price of the security over some future time period and then look for a call option with a strike price and premium in that time period that will allow them to exercise the option in-the-money. Because the trader is buying a call option that gives them the right to purchase a security at some point in the future, the trader will only be able to to exercise the option in-the-money if the price of the underlying security rises. This is because when the option is exercised by the trader, they purchase the underlying security for the strike price and then sell it for whatever price the security is currently trading for on the open market. If the price of the security is lower than the strike price when the option is exercised, the investor would lose money. In this case, the investor would not exercise the option and would effectively lose whatever premium they paid for the option in the first place. Formally, the option payoff is expressed as $[S-K]^+$, where this is defined to be:

$$[S-K]^+ = Max[(S-K),0] \tag{1.8}$$

where S is the price of the security and K is the strike price. If a trader sees an attractive security price and would like to lock it in without a large initial investment, they can buy a call option which is usually much less than the cost of the security. As explained in the previous section, if the option can be exercised in-the-money, the trader will do so, and if not, the most the trader loses is the cost of the option. A put option is very similar, but instead, the owner has the right to sell the underlying security at the strike price prior to the expiry date. With a put option, if the underlying security drops below the strike price, the owner of the option can purchase the underlying security on the market, exercise the option, and then sell the security for a payoff of $[K - S]^+$. If the option expires or the underlying security never falls below the strike price, the option becomes worthless. Call options and put options are the two types of options, but there are many styles of options with varying underlying financial instruments and payoff formulas. Several vanilla option styles and the exotic option styles that are the focus of this research are described below. For a more detailed discussion on various financial instruments please refer to [20].

European and American Options European and American options are the most basic styles of options and are typically referred to as vanilla options. The only difference between the two styles of options is American options can be exercised any time before or on the expiry date whereas European options can only be exercised on the expiry date. As described above, the payoff of European and American call options take the form $[S - K]^+$.

Asian Options An Asian option is also known as an average option. This is because their settlement value is based on either the arithmetic or geometric average value of the underlying financial instrument between a range of dates during the lifetime of the option. The Asian option payoff typically takes one of two forms, however, both forms settle European style. The first form is an *average price option* where the payoff is based on the difference between a fixed strike price and the average of the underlying financial instrument over the lifetime of the option. The other form is an *average strike option* where the payoff is determined like a regular option except the strike price is based on the average price of the underlying financial instrument over the lifetime of the option. The payoffs of the Asian call options take the form $[S_{avg} - K]^+$ and $[S - K_{avg}]^+$ respectively.

Bermudan Options A Bermudan option is a hybrid of American and European options. Bermudan options can be exercised early like American options, but can only be exercised early during certain date ranges over the lifetime of the option. Thus, prior to the early exercise date range the option behaves like a European option and after the early exercise date range it behaves like an American option. Bermudan options are typically used by companies as a form of compensation for senior executives joining the company as it motivates them to consider the long term benefits of their actions to the company.

Barrier Options Barrier options are very similar to regular options except that the option to exercise is based on the underlying financial instrument reaching a barrier price level. Barrier options typically take two forms, either *knock-out* or *knock-in*. As a result, they generally act like vanilla American or European options until one of the barrier events such as knock-out or knock-in occur. With knock-out, if the underlying financial instruments reach a specific barrier price level, the option terminates and expires worthless. With knock-in, the option does not become effective until the underlying financial instruments reach a certain price, and if they don't, the option expires worthless. Considering there are more conditions that must be met in order to receive a payoff and that the payoff is at best as good as a regular option, the premium on Barrier options is usually less. Four specific styles of Barrier options are Up-and-out, Down-and-out, Up-and-in and Down-and-in. Up-and-out Barrier options are where the spot price starts below the barrier level and has to move up for the option to expire worthless. Down-and-out Barrier options are where the spot price starts above the barrier level and has to move down for the option to expire worthless. Up-and-in Barrier options are where spot price starts below the barrier level and has to move up for the option to become effective. Down-and-in Barrier option are where spot price starts above the barrier level and has to move down for the option to become effective.

Pricing Options

Determining the price (value) of an option is central to financial instrument valuation theory and is the key theme of this research. Several models have been proposed to price options but the most popular, and the one awarded the 1997 Nobel Prize in Economics, is the Black-Scholes-Merton pricing model. By assuming the *efficient market hypothesis* [5] is true, i.e. that no arbitrage (risk-free profit) opportunities should exist in a market where options are correctly priced, Black, Scholes and Merton developed a formula for pricing options [7]. Their research examines previous attempts to develop a formula for pricing options, and concludes that each of them are flawed, as they all contain at least one parameter that is either inappropriate or effectively guessed. In order to develop their options pricing formula, they first make the following assumptions:

1. There are no arbitrage opportunities.

2. The short term interest rate is known and constant through time.
3. The stock price follows a random walk in continuous time with a variance rate proportional to the square of the stock price.
4. The distribution of stock prices is lognormal.
5. The stock pays no dividends or other distributions.
6. The option is European.
7. There are no transaction costs or taxes.
8. It is possible to borrow and lend cash at a constant risk-free interest rate.
9. It is possible to short sell the underlying stock without penalty.
10. All securities are perfectly divisible.

With these assumptions it is possible to create a perfectly hedged position with a long position in the stock and a short position in the option such that the value of the option is dependant only on time and known constant values. This portfolio is created by taking a short position equal to $1/(\frac{\partial V}{\partial S})$ options where $V(S,t)$ is the value of the option in relation to the stock price S and time t. If this hedge is continuously maintained, then the expected return from the portfolio becomes the risk-free interest rate as the portfolio has no risk of losing money. If this were not true, arbitrage would be possible. The result is a partial differential equation for the price of the option as follows:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{1.9}$$

Noting that the boundary condition of a European call option is $C(S,t) = [S-K]^+$, the solution to this differential equation is the following:

$$C(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2) \tag{1.10}$$

$$d_1 = \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \tag{1.11}$$

$$d_2 = \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \tag{1.12}$$

where $C(S,t)$ is the call option price, $N(d)$ is the cumulative normal density function (which is the most difficult of the variables to calculate in practice), r is the risk-free interest rate, $(T-t)$ is the maturity time and σ^2 is the variance rate. The key component of this result (which many find surprising) is that the return on the stock is not a variable in the formula for the option price.

The Black-Sholes-Merton valuation approach can be extended to other styles of options. In particular, American call options can be valued the same way because it has been shown that they have the same value as a European call option [26]. This was proven by showing that a rational investor would continue to hold an American option until maturity, even if it is in-the-money. Further, the pricing formula can be extended to European put options by simply changing the boundary condition to be $P(S,t) = [K-S]^+$ where $P(S,t)$ is the value of a put option. Further calculation shows a European put price can be expressed as follows:

$$P(S,t) = -SN(-d_1) + Ke^{-r(T-t)}N(-d_2) \tag{1.13}$$

Unfortunately, no analytical formula for an American put option has been derived, which has given rise to the popularity of numerical methods.

Chapter 2

Related Work

There is a large body of research on classical option pricing and while there has been some research in the area of quantum finance, there has been comparatively little research in the area of quantum option pricing. Most quantum option pricing research so far has typically focused on the quantization of the classical Black-Scholes-Merton equation from the perspective of continuous equations like the Schrödinger equation. On the other hand, metaphorically speaking, Chen's quantum binomial model (referred to hereafter as the quantum binomial model) is to existing quantum finance models what the Cox-Ross-Rubinstein model was to the Black-Scholes-Merton model — a discretized and simpler version of the same result. These simplifications make the respective theories not only easier to analyze but also easier to implement on a computer. This is crucial given that many option types and styles have no known analytical equation. This section will briefly outline research in quantum finance related to option pricing, with the exception being a detailed account of the Cox-Ross-Rubinstein classical binomial model and Chen's quantum binomial model.

Haven [18, 19] builds on the work of Chen and others but considers the market from the perspective of the Schrödinger equation. The key message in Haven's work is that the Black-Scholes-Merton equation is really a special case of the Schrödinger equation where markets are assumed to be efficient. The Schrödinger-based equation that Haven derives has a parameter $\hbar*$ (not to be confused with the complex conjugate of $\hbar$) that represents the amount of arbitrage that is present in the market resulting from a variety of sources including non-infinitely fast price changes, non-infinitely fast information dissemination and unequal wealth among traders. Haven argues that by setting this value appropriately, a more accurate option price can be derived, because in reality, markets are not truly efficient. This is one of the reasons why it is possible that a quantum option pricing model could be more accurate than a classical one.

Baaquie [5] has published many papers on quantum finance and even written a book [4] that brings many of them together. Core to Baaquie's research and others like Matacz [25] are Feynman's path integrals. Baaquie applies path integrals to several exotic options and presents analytical results comparing his results to the results of Black-Scholes-Merton equation showing that they are very similar. Baaquie is explicit in his book, "No attempt is made to apply quantum theory in re-working the fundamental principles of finance". Essentially, he reformulates classical results into quantum field theory language to obtain equivalent results. The results provide useful tools in this context, but they do not show how quantum computers could be leveraged, and an alternative finance theory based on quantum theory is not explored.

Piotrowski et al. [31] take a different approach by changing the Black-Scholes-Merton assumption regarding the behavior of the stock underlying the option. Instead of assuming it follows a Wiener-Bachelier [20] process, they assume that it follows a Ornstein-Uhlenbeck [34] process. With this new assumption in place, they derive a quantum finance model as well as a European call option formula. Other models such as Hull-White [21] and Cox-Ingersoll-Ross [11] have successfully used the same approach in the classical setting with interest rate derivatives. That said, Piotrowski et al. do not provide any quantitative results or analysis regarding the impact of using the Ornstein-Uhlenbeck process for option prices in general.

Khrennikov [23] builds on the work of Haven and others and further bolsters the idea that the market efficiency assumption made by the Black-Scholes-Merton equation may not be appropriate. To support this idea, Khrennikov builds on a framework of *contextual probabilities* using agents as a way of overcoming criticism of applying quantum theory to finance. Although the work is interesting, no option pricing formula was derived.

Accardi and Boukas [2] again quantize the Black-Scholes-Merton equation, but in this case, they also consider the underlying stock to have both Brownian and Poisson processes. Again, no quantitative results or analysis regarding the impact of their choice is provided.

Almost all of the research examined in this area has been completely theoretical and lacked quantitative results with analysis. This is one of the motivations for simulating option pricing algorithms as part of this research.

2.1 Classical versus Quantum Binomial Model

For simplicity and clarity, the following sections show the derivation of both the classical and quantum binomial models assuming only a single step of the simulation will be run. This is then followed by an extrapolation of the single step quantum model to the multi-step quantum binomial model. Both the quantum and classical binomial model derivations use one key common assumption to price an option. They assume an arbitrage-free replicating portfolio instead of the alternative, yet equivalent, arbitrage-free delta hedging portfolio or risk-neutral valuation assumption. Considering the arbitrage-free replicating portfolio technique is used frequently below, it is described here as follows. First of all, a portfolio is created that consists of Δ shares of a stock and B dollars borrowed at the risk-free rate which, is typically assumed to be the London Interbank Offered Rate (LIBOR) in practice. Δ and B are selected such that they emulate the cash flows of the option to be valued. The replicating portfolio technique assumes that there should never be any arbitrage opportunities and, as a result, the replicating portfolio should have the same value as the option. If this were not true, the portfolio could be sold and the option could be purchased for a risk-less (arbitrage) profit. Consider the following call option example where K is the strike price, S_u is the stock price as it increases in value and S_d is the stock price as it decreases in value:

$$\Delta S_u - B(1+r) = [S_u - K]^+ \tag{2.1}$$

$$\Delta S_d - B(1+r) = [S_d - K]^+ \tag{2.2}$$

After solving for both Δ and B the value of the call option C is:

$$C = \Delta S_c - B \tag{2.3}$$

where S_c is the current stock price.

As mentioned previously, another key consideration when comparing the classical versus quantum model is the statistical distribution chosen. If the Maxwell-Boltzmann classical statistical distribution is used with the quantum binomial model the price will match the classical price and the model will collapse to the Black-Scholes-Merton model. Alternatively, if a quantum statistical distribution such as Fermi-Dirac or Bose-Einstein is used, the price will not be the same as the Black-Scholes-Merton model and instead, will be equal to what will be referred to as the quantum binomial model price.

2.1.1 Derivation of the Single-Step Classical Binomial Model

The classical binomial stock pricing model derived by Cox-Ross-Rubinstein translates the Black-Scholes-Merton model into a discrete binary tree of prices. Essentially, it is a numerical method for the valuation of options that is very popular because of its flexibility and for its ability to be executed on a computer. For each n steps of the binomial model, n new tree nodes are created. These new nodes represent a single discrete change in the underlying stock price. The execution of each step increases the accuracy of the resulting option price, eventually converging to the Black-Scholes-Merton price. The term single-step means that only one discrete change in the stock underlying the option will be considered as shown in Figure 2.1. With this, the classi-

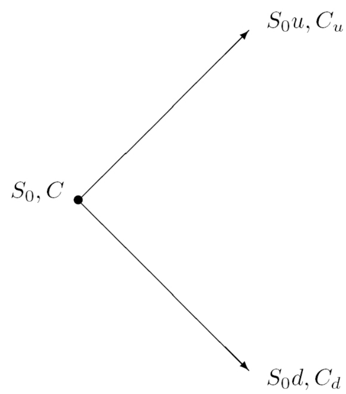


Figure 2.1: Single Step Classical Binomial Model

cal binomial stock pricing model is derived as follows. With B representing a risk-free bank account and S representing a stock, an arbitrage-free replicating portfolio can be set-up as follows:

$$B_1 = B_0(1 + r),\ S_1 = S_0(1 + R) \tag{2.4}$$

where R is random variable taking just two values, causing S to go either up or down. C_u is the price of the call option if there is an upward movement u in the stock price[1]:

$$C_u = [S_0u - K]^+ \tag{2.5}$$

C_d is the price of the call option if there is a downward movement d in the stock price:

$$C_d = [S_0d - K]^+ \tag{2.6}$$

where, by definition, $u \geq 1$ and $0 < d \leq 1$. In order to account for the time value of money and that the Black-Scholes-Merton model is continuous, the present value

[1]Recall that $[S_0 - K]^+$ is shorthand for $Max[S_0 - K, 0]$

of C at time 0 is derived by discounting the future value of C using the continuous compounding interest rate formula [20] as follows:

$$C = e^{-rT}[qC_u + (1-q)C_d] \tag{2.7}$$

Because of the form of equation 2.7, q is interpreted as the probability of an upward stock movement, and $1-q$ is interpreted as the probability of a downward stock movement in a risk-free world where:

$$q = \frac{e^{rT} - d}{u - d} \tag{2.8}$$

Recalling that the distribution of the stock price is lognormal, u and d are estimated as a function of the historical stock volatility σ as follows:

$$u = e^{\sigma\sqrt{\Delta t}} \tag{2.9}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{2.10}$$

Substituting for q in equation 2.7 we get the expanded formula for the price of an option C:

$$C = e^{-rT}\left[\frac{e^{rT} - d}{u - d}C_u + \left(1 - \frac{e^{rT} - d}{u - d}\right)C_d\right] \tag{2.11}$$

2.1.2 Derivation of the Single-Step Quantum Binomial Model

The following is a brief outline of the derivation of the quantum binomial method based on the work of Chen [10]. First, the quantum mechanical foundation for a quantum-based no-arbitrage stock market is presented, which is followed by the quantization of the classical binomial model. Assume a stock is in a quantum state:

$$\rho = \frac{1}{2}(wI_2 + x\sigma_x + y\sigma_y + z\sigma_z) \tag{2.12}$$

where ρ is an arbitrary 2×2 Hermitian matrix[2] and is known as the density matrix. In quantum mechanics, a density matrix is, by definition, a Hermitian matrix of trace one. Density matrices are used to describe the statistical state of either an ensemble of systems or a single system where the pure quantum state the system is in, is unknown. Note that any 2×2 Hermitian matrix can be written as a linear combination of the Pauli matrices[3], as they form the basis for the Hilbert space of 2×2 complex Hermitian matrices. In addition, consider a matrix A (known as a quantum operator in quantum mechanics) that is used to transform the stock from one state to the next:

$$A = (x_0 I_2 + x_1\sigma_x + x_2\sigma_y + x_3\sigma_z) \tag{2.13}$$

where A is an arbitrary Hermitian matrix. Let a and b be the eigenvalues of A, which means they represent the possible values A can take when it is measured. Eigenvalues

[2] A Hermitian matrix is a complex square matrix that is equal to its own conjugate transpose.
[3] The Pauli matrices:

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

a and b can be expressed as follows:

$$a = x_0 - \sqrt{x_1^2 + x_2^2 + x_3^2},\ b = x_0 + \sqrt{x_1^2 + x_2^2 + x_3^2} \tag{2.14}$$

such that all x_j are real numbers and that $x_1^2 + x_2^2 + x_3^2 \neq 0$ and $a, b > -1$. After solving each equation for x_0 and setting them equal to each other, the following result is obtained:

$$a + \sqrt{x_1^2 + x_2^2 + x_3^2} = b - \sqrt{x_1^2 + x_2^2 + x_3^2} \tag{2.15}$$

which can be rewritten as:

$$\frac{(b-a)^2}{4} = x_1^2 + x_2^2 + x_3^2 \tag{2.16}$$

By substituting $(b-a)/2$ into equation 2.14 in place of $\sqrt{x_1^2 + x_2^2 + x_3^2}$, it follows that:

$$x_0 = a + \frac{b-a}{2} = \frac{a+b}{2} \tag{2.17}$$

Black-Scholes-Merton derivative pricing theory says that in a risk-neutral world, one should expect to earn the risk-free interest rate. Considering A is used to transform the stock from one state to the next, one should expect that it evolves the stock at the risk-free rate. Thus, the expected value of measuring A should be the risk-free interest rate. In quantum mechanics, the expected value of a quantum operator can be calculated as $\langle A \rangle_\rho = \mathrm{tr}(\rho A)$, where tr is the trace matrix operation, and ρ is the density matrix. As a result:

$$\langle A \rangle_\rho = \mathrm{tr}(\rho A) = r \tag{2.18}$$

where r is the risk-free rate. As discussed earlier, a density matrix must have a trace of one:

$$\mathrm{tr}\rho = \frac{1}{2}(w+z) + \frac{1}{2}(w-z) = 1 \tag{2.19}$$

and as a result $w = 1$, which means that:

$$r = \mathrm{tr}\left(\frac{1}{2}\begin{pmatrix} w+z & x-iy \\ x+iy & w-z \end{pmatrix}\begin{pmatrix} x_0+x_3 & x_1-x_2i \\ x_1+x_2i & x_0-x_3 \end{pmatrix}\right)$$

which reduces to:

$$r = wx_0 + zx_3 + xx_1 + x_2y \tag{2.20}$$

After substituting $x_0 = (a+b)/2$ and $w = 1$ into equation 2.20, the risk-neutral states are shown to be:

$$x_1x + x_2y + x_3z = r - \frac{a+b}{2} \tag{2.21}$$

Considering, by definition, the eigenvalues of any density matrix must be $0 \leq \lambda_i \leq 1$, the eigenvalues of ρ are:

$$\lambda_1 = \frac{1}{2}(w - \sqrt{x^2+y^2+z^2}),\ \lambda_2 = \frac{1}{2}(w + \sqrt{x^2+y^2+z^2}) \tag{2.22}$$

where $w = 1$, meaning the eigenvalues must have:

$$x^2 + y^2 + z^2 < 1 \tag{2.23}$$

This means that the geometry of the risk-neutral states defined by equations 2.22 and 2.23 is a disk with radius:

$$\sqrt{1 - \frac{(2r - a - b)^2}{(b - a)^2}} \tag{2.24}$$

in the unit ball of $\mathbf{R}^3$. The quantum binomial model replaces the single random variable R in the classical model with a complex Hermitian matrix A much like Heisenberg did in 1925 to quantize Newtonian Physics. The result is the following quantum single step binomial model for the price of a European call option. With B representing a risk-free bank account and S representing a stock, an arbitrage-free replicating portfolio can be setup as follows:

$$B_1 = B_0(1 + r),\ S_1 = S_0(I_2 + A) \tag{2.25}$$

where the quantum operator A is built as per equation 2.13. h_b is the price of the call option if there is an upward movement in the stock of $(1 + b)$:

$$h_b = [S_0(1 + b) - K]^+ \tag{2.26}$$

h_a is the price of the call option if there is a downward movement in the stock of $(1 + a)$:

$$h_a = [S_0(1 + a) - K]^+ \tag{2.27}$$

C is the price of the European call option at time 0 discounted by $1/(1+r)$ assuming the market is quantum:

$$C = \frac{1}{1+r}\left[\left(\frac{b-r}{b-a}\right)h_a + \left(\frac{r-a}{b-a}\right)h_b\right] \tag{2.28}$$

2.1.3 Multi-Step Quantum Binomial Model

The following equations take the single-step quantum binomial model and extrapolate it to an N-period multi-step model. In the multi-step model, each step is tensored[4] with the previous step to build a composite quantum system that represents the entire history of the simulation of the option price. With B representing a risk-free bank account and S representing a stock, an arbitrage-free replicating portfolio can be set-up as follows:

$$B_1 = B_0(1+r)^n,\ S_n = S_0 \bigotimes_{j=1}^{n}(I_2 + A_j) \otimes I_{N-n} \tag{2.29}$$

where the quantum operator A is a complex Hermitian matrix representing stock price movement and is built as follows:

$$A_j = x_0 I_2 + x_{1j}\sigma_x + x_{2j}\sigma_y + x_{3j}\sigma_z \tag{2.30}$$

where $\sigma_x, \sigma_y, \sigma_z$ are the Pauli spin matrices of quantum mechanics, for all $j = 1, ..., N$. S_N can be represented as follows, assuming the Maxwell-Boltzmann classical statis-

[4]Recall that $\otimes$ represents the tensor product (otherwise known as the Kronecker product).

tics:

$$S_N = S_0 \sum_{n=0}^{N} (1+b)^n (1+a)^{N-n} \left[\sum_{|\sigma|=n} \bigotimes_{j=1}^{N} |w_{j\sigma}\rangle \langle w_{j\sigma}| \right] \tag{2.31}$$

where all σ (not to be confused with stock volatility) are subsets of $\{1, ..., N\}$, $w_{j\sigma} = u_\sigma$ for $j \in \sigma$ or $w_{j\sigma} = v_\sigma$ otherwise and form an orthonormal basis in the Hilbert space. With these definitions, $[S_N - K]^+$ can now be represented as follows:

$$[S_N - K]^+ = \sum_{n=0}^{N} [S_0 (1+b)^n (1+a)^{N-n} - K]^+ \left[\sum_{|\sigma|=n} \bigotimes_{j=1}^{N} |w_{j\sigma}\rangle \langle w_{j\sigma}| \right] \tag{2.32}$$

The density matrix representing the stock's quantum state ρ is constructed as follows:

$$\bigotimes_{j=1}^{N} \rho = \frac{1}{2^N} \bigotimes_{j=1}^{N} (I_2 + x_j \sigma_x + y_j \sigma_y + z_j \sigma_z) \tag{2.33}$$

where each j will require an additional qubit in order to be simulated with a quantum computer. The resulting option price is then:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[S_N - K]^+] \tag{2.34}$$

Chen then takes this equation and derives the equivalent of the Cox-Ross-Rubinstein option pricing formula as follows:

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [S_0 (1+b)^n (1+a)^{N-n} - K]^+ \tag{2.35}$$

This shows that assuming stocks behave according to Maxwell-Boltzmann classical statistics, the quantum binomial model does indeed collapse to the classical binomial model. Chen also shows that classical Maxwell-Boltzmann statistics can be replaced by the quantum Bose-Einstein statistics resulting in the following option price formula:

$$C_0^N = \mathrm{tr}[(\rho^{\hat{\bigotimes} N})[S_N - K]^+] \tag{2.36}$$

Chen then takes this equation and derives a new quantum option pricing formula that is not the equivalent of the Cox-Ross-Rubinstein as follows:

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \left(\frac{q^n (1-q)^{N-n}}{\sum_{k=0}^{N} q^k (1-q)^{N-k}} \right) [S_0 (1+b)^n (1+a)^{N-n} - K]^+ \tag{2.37}$$

Equation 2.37 will produce option prices that will differ from those produced by the Cox-Ross-Rubinstein option pricing formula in certain circumstances. This is because the stock is being treated like a quantum boson particle instead of a classical particle.

Chapter 3

Results

The results of this thesis research are described in the following sections. First of all, the classical and quantum binomial models are compared and contrasted to each other. Secondly, to give the appropriate context to the discussions relating to extending the model, the base quantum binomial model simulation is presented. Finally, the extension of the quantum binomial model to styles other than European call options is presented and evaluated.

3.1 Differences Between the Classical and Quantum Binomial Model

The differences between the classical and quantum binomial models are summarized in Table 3.1. If one took the approach of J. Bernoulli and C. Huygens [6] for calculating the expected value of a random variable in the 1700's, an option could be

CV	QV	Classical Meaning	Quantum Meaning
B_n	B_n	Risk Free Bank Account after Step n	Risk Free Bank Account after Step n
S_n	S_n	Stock Price at Step n	Stock Price at Step n
K	K	Strike Price	Strike Price
r	r	Risk-Free Rate	Risk-Free Rate
R	A	Random Real Scalar Growth	Random Complex Unitary Matrix Growth
d	1+a	Downward Movement Amount	Downward Movement Amount
u	1+b	Upward Movement Amount	Upward Movement Amount
C_d	h_a	Downward Option Price	Downward Option Price
C_u	h_b	Upward Option Price	Upward Option Price
q	q	Upward Stock Movement Probability	Upward Stock Movement Probability
σ	σ	Stock Volatility	Stock Volatility
Δt	Δt	Time Step	Time Step
T	T	Total Time	Total Time

Table 3.1: Comparison of classical variables (CV) and quantum variables (QV)

priced as follows:

$$C = \frac{1}{1+r} E(S_1 - K)^+ = \frac{1}{1+r}(qh_b + (1-q)h_a) \tag{3.1}$$

where q is the probability that the random variable R takes the value h_b. Equation 3.1 is in the same form as the classical binomial equation 2.7 above, but q isn't estimated using volatility σ like in equations 2.9 and 2.10. The problem with this is that the value of the option is based on some arbitrary assumption of the probability q of the stock moving up or down. This is precisely the problem that the Black-Scholes-Merton equation solved by developing a formula that prices options only on the stock's volatility and not its expected return. That said, the Black-Scholes-Merton approach is effectively estimating the probability of stock movement using historical volatility and thus not entirely removing it from the equation. The quantum binomial model represents probability with density matrices and when repeated measurements of the quantum system are made, the expected value of the quantum operator A is calculated which equals the price of the option. This is accomplished as follows, stock movement is based on the evolution of the quantum rate of return A until a measurement is made. When a measurement is finally made, the equation collapses to the classical binomial model with the option value being based on the eigenvalues of A as shown in equation 2.28. Each eigenvalue a and b has a specific probability of being the value returned during the measurement. The following section describes how the probability of measuring a specific eigenvalue is calculated. First of all, assume u, v form an orthonormal basis in $\mathbf{C}^2$. A can then be written in terms of outer products of the basis and its eigenvalues as follows:

$$A = a \left|u\right\rangle \left\langle u\right| + b \left|v\right\rangle \left\langle v\right| \tag{3.2}$$

where a and b are the eigenvalues of A and $\left|u\right\rangle \left\langle u\right|$ means the outer product of the vectors $\left|u\right\rangle$ and $\left\langle u\right|$. If the orthonormal basis is $\left|u\right\rangle = (1,0)^T$ and $\left|v\right\rangle = (0,1)^T$ then the outer product $\left|u\right\rangle \left\langle u\right|$, is a 2×2 matrix with a 1 in position $[1,1]$ and 0's elsewhere as shown in equation 1.6. In quantum mechanics, the probability that measuring a system in the state ρ will result in the eigenvalue a is $\left\langle u\right| \rho \left|u\right\rangle$ and the probability it will result in the eigenvalue b is $\left\langle v\right| \rho \left|v\right\rangle$. Thus, the probability that A takes the value a or b after measurement can be expressed as follows:

$$p(a) = \left\langle u\right| \rho \left|u\right\rangle = \frac{1}{2}[u_1 u_1^*(w+z) + u_1^* u_2(x-iy) + u_2^* u_1(x+iy) + u_2^* u_2(w-z)] \tag{3.3}$$

$$p(b) = \left\langle v\right| \rho \left|v\right\rangle = \frac{1}{2}[v_1 v_1^*(w+z) + v_1^* v_2(x-iy) + v_2^* v_1(x+iy) + v_2^* v_2(w-z)] \tag{3.4}$$

which reduces to:

$$p(a) = \langle u| \, \rho \, |u\rangle = \frac{1}{2}[(w + z)] \tag{3.5}$$

$$p(b) = \langle v| \, \rho \, |v\rangle = \frac{1}{2}[(w - z)] \tag{3.6}$$

The quantum binomial model is just like any quantum model in that, in order for it to evolve from state to state and still remain in a quantum state, it must not be measured. Thus measurement of the quantum binomial model should not occur until it has reached the desired number of steps required for the desired accuracy of the option price. This means that during the evolution of the stock from one quantum state to the next without measurement, the probability of the stock moving up or down is not manifested and the the density matrix ρ preserves the entire statistical ensemble of the state. Essentially, in the classical binomial model it is as if a measurement is occurring after each step of the model. In this case, it would be rational for the measuring observer to think that the value of their option is based on the probability of the stock going up or down. The irony is that the observer is the one causing probability to become a factor as they are collapsing the quantum state by performing the measurement. The fact that the classical form can be re-written in probabilistic form is what Chen is calling the *paradox*. This is because, the probability of the up and down option prices is already built into the movement of the underlying stock prices, thus, it seems *paradoxical* that probability would re-enter the equation.

3.2 Base Algorithm for Simulating the Quantum Binomial Model

European call options, European put options and Barrier options were simulated as part of this research. The base algorithm for simulating these options, developed as part of this research, has four main phases.

Derive the Eigenvectors of the Quantum Operator The first phase is to derive the eigenvectors that will be used as the canonical basis for the simulation for a given volatility σ and a number of periods N. Deriving the eigenvectors has several steps as follows. Using the formula originally proposed by Meyer [27] and corrected as part of this research, for volatility:

$$\sigma = \frac{\ln\left(1 + x_0 + \sqrt{x_1^2 + x_2^2 + x_3^2}\right)}{\sqrt{1/t}} \tag{3.7}$$

that can be rewritten in terms of b by substituting in equation 2.14 as follows:

$$\sigma = \frac{\ln(1 + b)}{\sqrt{1/t}} \tag{3.8}$$

where $t = T/N$ is the time of each period. The eigenvalue b can then be calculated by rearranging equation 3.8 as follows:

$$b = e^{\sigma\sqrt{1/t}} - 1 \tag{3.9}$$

Using the value of b and the risk free rate r as constraints, the rest of the parameters for A_0 and ρ shown in equations 2.13 and 2.12 respectively are chosen such that the risk free equation 2.24 is satisfied. With these parameters chosen, the eigenvalue a is calculated as per equation 2.14 and the operator A_0 for period 1 is constructed as per equation 2.13. Next, the eigenvectors v and u of A_0 are derived using both the eigenvalues a and b. The eigenvectors, v and u are then used as the canonical basis for further calculations. Note that the parameters x_1, x_2, x_3 that drive the operator A can be adjusted as per equation 3.7 to achieve the desired volatility σ for the simulation.

Initialize the Input Values The second phase is the preparation of the input values $|\psi_0\psi_1\psi_2\psi_3...\psi_N\rangle$ according to the density matrix ρ as follows:

$$\rho = \bigotimes_{j=1}^{N}(|u\rangle\langle u|(1-q) + |v\rangle\langle v| q) \tag{3.10}$$

where q is derived as per equation 2.8.

Build the Quantum Operator The third phase is to build the quantum operator A that will be used to evolve the stock price S through the N periods. This is done by combining the eigenvectors v and u, which form the canonical basis of the vector space, as follows, from equation 2.31:

$$\sum_{|\sigma|=n} \bigotimes_{j=1}^{N} |w_{j\sigma}\rangle\langle w_{j\sigma}| \tag{3.11}$$

where all σ (not to be confused with volatility) are subsets of $\{1, ..., N\}$, $w_{j\sigma} = u_\sigma$ for $j \in \sigma$ or $w_{j\sigma} = v_\sigma$ otherwise. Note that m will be used to represent the number of subsets per period as follows:

$$m = |\sigma| = \frac{N!}{n!(N-n)!} \tag{3.12}$$

which can also be represented by Pascal's triangle. There are several steps required to combine the subsets in accordance with equation 3.11. First of all, the subset number m is converted to binary for each subset. Secondly, subsets with the same number of 0s in their binary subset numbers are grouped together. Thirdly, the groups are sorted in descending order of the number 0s in their respective binary subset number m. Finally, for each binary subset number m, each 0 is set to $|u\rangle\langle u|$, each 1 is set to $|v\rangle\langle v|$ and then the tensor product of each of the resulting outer products is calculated[1]. This is illustrated with following examples where n is the current period and m is the number subsets within the period. When $N = 1$,

$$n_0 = |u\rangle\langle u| + |v\rangle\langle v| \tag{3.13}$$

When $N = 2$,

$$\begin{aligned} n_0 &= m_0 = |u\rangle\langle u| \otimes |u\rangle\langle u| \\ n_1 &= m_1 + m_2 = (|u\rangle\langle u| \otimes |v\rangle\langle v|) + (|v\rangle\langle v| \otimes |u\rangle\langle u|) \\ n_2 &= m_3 = |v\rangle\langle v| \otimes |v\rangle\langle v| \end{aligned} \tag{3.14}$$

[1]For example, $9 = 1001 = |v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v|$

When $N = 3$,

$$
\begin{aligned}
&n_0 = m_0 = |u\rangle \langle u| \otimes |u\rangle \langle u| \otimes |u\rangle \langle u| \\
&n_1 = m_1 + m_2 + m_4 = (|u\rangle \langle u| \otimes |u\rangle \langle u| \otimes |v\rangle \langle v|) + (|u\rangle \langle u| \otimes |v\rangle \langle v| \otimes |u\rangle \langle u|) \\
&+ (|v\rangle \langle v| \otimes |u\rangle \langle u| \otimes |u\rangle \langle u|) \\
&n_2 = m_3 + m_5 + m_6 = (|u\rangle \langle u| \otimes |v\rangle \langle v| \otimes |v\rangle \langle v|) + (|v\rangle \langle v| \otimes |u\rangle \langle u| \otimes |v\rangle \langle v|) \\
&+ (|v\rangle \langle v| \otimes |v\rangle \langle v| \otimes |u\rangle \langle u|) \\
&n_3 = m_7 = |v\rangle \langle v| \otimes |v\rangle \langle v| \otimes |v\rangle \langle v|
\end{aligned}
\tag{3.15}
$$

When $N = 4$,

$$
\begin{aligned}
&n_0 = m_0 = |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \\
&n_1 = m_1 + m_2 + m_4 + m_8 \;= (|u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v|) \\
&+ (|u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u|) + (|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u|) \\
&+ (|v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u|) \\
&n_2 = m_3 + m_5 + m_6 + m_9 + m_{10} + m_{12} = (|u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v|) \\
&+ (|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v|) + (|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u|) \\
&+ (|v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v|) + (|v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u|) \\
&+ (|v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |u\rangle\langle u|) \\
&n_3 = m_7 + m_{11} + m_{13} + m_{14} = (|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v|) \\
&+ (|v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v|) + (|v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u| \otimes |v\rangle\langle v|) \\
&+ (|v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |u\rangle\langle u|) \\
&n_4 = m_{15} = |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes |v\rangle\langle v|
\end{aligned}
\tag{3.16}
$$

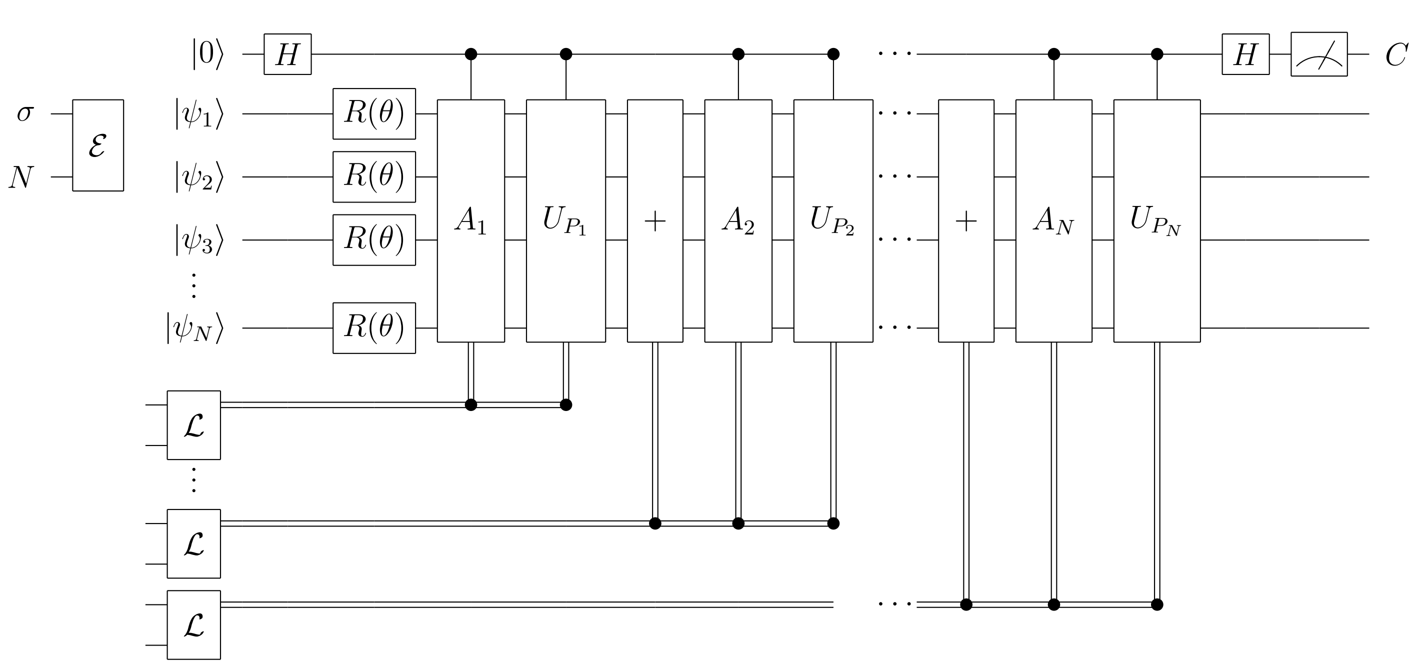
Calculate the Payoff Formula for the Option Style In this phase, the payoff formula for the option style being simulated is calculated and it involves several steps as follows. For each n, the stock price $S_N = S_{N-1} + S_n$ is calculated (S_n is equivalent to the value of the n^{th} leaf node of the Cox-Ross-Rubinstein binomial tree) using equation 2.31. The next step is to implement the payoff formula and if the result is non-zero, multiply it by the quantum operator matrix created in the previous phase. For example, for a European call option, where $N = 2$ and $n = 1$, this step would perform the following calculation:

$$S_n = [S_0(1+b)^n(1+a)^{N-n} - K]^+(|u\rangle\langle u| \otimes |v\rangle\langle v| + |v\rangle\langle v| \otimes |u\rangle\langle u|) \tag{3.17}$$

Calculate the Expected Value of the Quantum Operator The final phase is to determine the option price C_N by calculating the expected value of the quantum operator S_N as per equation 2.34.

3.2.1 Multi-Step Quantum Binomial Algorithm Quantum Circuit

The following circuit implements the multi-step quantum binomial algorithm described above:



3.2.2 Complexity Analysis of the Quantum and Classical Binomial Algorithms

The first part of the quantum circuit above is the $\mathcal{E}$ function call, which is a classical routine, that derives the eigenvectors of the quantum operator based on volatility σ and the number of periods N. The amount of processing required for $\mathcal{E}$ is constant as it requires four elementary calculations and solving for the eigenvectors of a 2×2 matrix. The second part of the circuit is the preparation of the input values $|\psi_0\psi_1\psi_2\psi_3...\psi_N\rangle$ according to the density matrix ρ in equation 3.17 which can be done in one step [28], using N rotation gates[2] $R_y(\theta)$ with appropriately selected values of θ. In the third part of the circuit, each quantum operator A_n is built as per the subset algorithm described in the section above which takes $O(2^N)$ time. Note that the overhead to create the operators is borne while the circuit is being built and not during the execution of the circuit. This means that subsequent simulations, where values other than volatility σ and the number of periods N change, can be run without incurring the overhead associated with building the operators. However, it also means that quantum binomial option pricing with stochastic volatility will incur this overhead every time. The next part of the circuit implements the payoff function

[2]
$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

for the option style being simulated. $\mathcal{L}$ contains the payoff condition for the option style being simulated and if the payoff is positive, the associated quantum operator A_n times $U_{P_n} = (S_0(1+b)^n(1+a)^{N-n} - K) \times I_N$ will act on the input qubits and add it to the value calculated in the previous step using the addition gate [14]. The addition gate has a runtime of $O(\log_2 N)$ and uses $2N$ qubits (the additional qubits are not shown in the quantum circuit diagram). The final step is to calculate the expected value of the quantum operator. This is done using the gates on the wire with the $|0\rangle$ ancillary qubit. The two Hadamard[3] gates H and the control wires, calculate the expected value of the A operator $\mathrm{tr}(\rho A)$ in constant time $O(1)$ which is equal to the final value of the option C. The time complexity of the entire circuit, ignoring constant time operations, is therefore $O(2^N + N\log_2 N)$. If the same circuit is used to run another simulation, but the volatility σ and the number of periods N are kept the same, the time complexity is $O(N\log_2 N)$. The circuit can be optimized by considering the fact that only periods where the option is in-the-money need to be executed. The option is in-the-money when $S_0(1+b)^n(1+a)^{N-n} > K$. The resulting time complexity with this optimization is $O((N-\tau)\log_2(N-\tau))$ where τ is the boundary where the option becomes in-the-money. Further, because the quantum circuit can be built using a polynomial number of gates $O(N-\tau)$ and it can be *uniformly generated*, it is in the quantum complexity class **BQP**.

For certain option styles, such as American put options, exercising the option prior to its expiry date may be more advantageous. This is know as *early exercise*.

[3]
$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Classical binomial option price algorithms that take early exercise into account, need to evaluate every node in the binomial tree and have a time and space complexity of $O(N^2)$ [24]. Classical binomial option price algorithms that do not take early exercise into account, are more efficient. One such algorithm [24] has a time complexity of $O(N - \tau)$, and a space complexity of $O(1)$ where τ is the boundary where the option becomes in-the-money. The quantum algorithm above does not take into account early exercise, however, it preserves all of the information that would be contained within a full binomial tree. So although it runs slower than a classical binomial algorithm that does not take into account early exercise, it does run faster than one that does. Further, because the matrix representing every period of the simulation is generated, it seems reasonable to expect that there should be a way to incorporate early exercise into the algorithm without much change in its complexity. Modifying the algorithm to consider early exercise has been left to future research.

3.3 Extending and Evaluating the Quantum Binomial Model

The following extends the quantum binomial model from European call options to European put options and to Barrier options by deriving the respective quantum mechanical binomial equations. In addition, challenges extending the quantum binomial model to American, Asian and Bermudan options while maintaining the ability to efficiently implement the model is analyzed. To accomplish this, the classical binomial models for each of the exotic option styles were analyzed. The analysis included determining what portions of the classical model need to be extended to the quantum model and which portions remain in their current classical form. Note that the analysis considered the most basic option pricing scenarios and does not consider the effect of dividends, variable interest rates, variable volatility or other complicating factors. Further, the analysis assumed that the payoff formula remained classical and that the stock evolution is quantum.

Several steps were taken in order to simulate the quantum binomial model. First of all, quantum circuits that implement the various options using the quantum binomial model were designed using the one described in the previous section as starting point. The second step was to implement the quantum circuit using a quantum circuit simulator. The quantum binomial model option prices were derived by coding the quantum circuits using the QuIDDPro quantum circuit simulator [35], whereas the classical values were simulated using DeriGem and tools from Hoadley Trading. Coding in QuIDDPro involved developing an I/O processing module, I/O file types, the quantum binomial model algorithm and a statistics module. In addition, the version of QuIDDPro used did not natively support factorial, exponentiation to a real number and natural logarithms. As a result, they had to be coded as they are each required to implement the quantum binomial algorithm. Further, the statistical distribution chosen for each simulation was the Maxwell-Boltzmann classical statistical distribution. Note that simulating Fermi-Dirac and Bose-Einstein statistics was left to future research. In general, with the available hardware, a 2.4GHz Linux server with 2GB of RAM, it was difficult to run simulations for more than 10 periods as the CPU usage was always at 100% (although the memory usage was typically around 2%). Another consideration when analyzing the results is the affect of rounding. DerivaGem is implemented in Excel so its precision is limited to 15 decimal places. The QuIDDPro simulations were also limited to 15 decimals in order to make the runtimes reasonable. Because the quantum binomial algorithm implemented in QuIDDPro has more steps than the classical one implemented in DerivaGem rounding has more of an impact on the final value that is calculated.

Each simulation was then evaluated using two key evaluation criteria. First is comparing the classical binomial model prices to the quantum binomial model prices. This was done by running simulations with judiciously-chosen scenarios and parameters to see how the resulting quantum option prices compare to classical option prices. Second is comparing the option pricing performance of the quantum binomial model to the classical binomial model. In order to compare the performance, complexity analysis was performed on the quantum binomial model in order to determine its efficiency for each option style.

European Options The payoff of European call options take the form $[S-K]^+$ whereas European put options take the form $[K-S]^+$. Therefore, the quantum binomial equation for the price of a European put option is:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[K - S_N]^+] \tag{3.18}$$

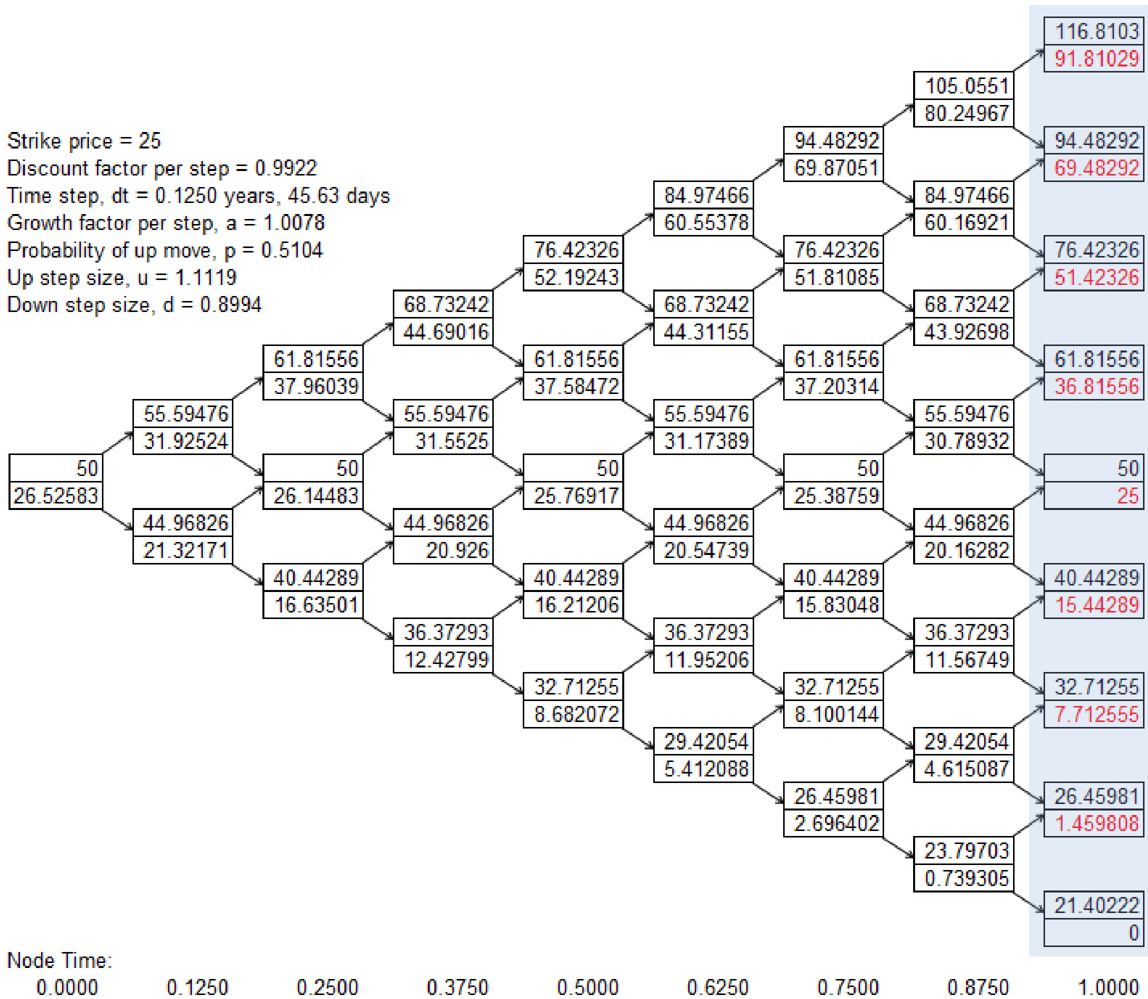


Figure 3.1: Eight period European call option classical binomial tree. The upper value in each box is the stock price S_n and the lower value is the option price C_n. $S_0 = 50$ and $K = 20$.

that can be expanded to the equivalent Cox-Ross-Rubinstein European put option pricing formula as follows:

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [K - S_0(1+b)^n(1+a)^{N-n}]^+ \quad (3.19)$$

Note that the values S_n and C_n calculated in each step of the simulation of the

Periods	2		3		8		9
S_n	C_n	S_n	C_n	S_n	C_n	S_n	C_n
32.7126	7.71255	29.7375	4.73747	21.4022	0	20.3285	0
50	25	42.0483	17.0483	26.4598	1.45981	24.8293	0
76.4233	51.4233	59.4555	34.4555	32.7126	7.71255	30.3265	5.32653
		84.069	59.069	40.4429	15.4429	37.0409	12.0409
				50	25	45.2419	20.2419
				61.8156	36.8156	55.2585	30.2585
				76.4233	51.4233	67.4929	42.4929
				94.4829	69.4829	82.4361	57.4361
				116.81	91.8103	100.688	75.6876
						122.98	97.9802
C_N	26.51467343		26.51467343		26.52583019		26.52429863

Table 3.2: The stock price S_n and the option price C_n for each step of various multi-period quantum binomial model simulations.

multi-period quantum binomial model equal the leaf nodes of the classical binomial model tree. This can be seen by comparing the S_n and C_n values found in the leaf nodes (surrounded by the blue box) in the classical binomial model in Figure 3.1 to the values in Table 3.2 for the 8 period quantum binomial simulation. Simulation of the classical binomial model using DerivaGem and the quantum binomial model using QuIDDPro can be seen in Figure 3.2. The simulation results are identical up to 9 decimals of precision with the difference attributed to rounding error.

American Options Just like European options, the payoff of American call options take the form $[S-K]^+$ whereas American put options take the form $[K-S]^+$. Without considering the effect of dividends, American call options will have the same price as European call options. On the other hand, American put options are not equivalent in price to European put options because it may be better to exercise an American option early. Considering early exercise is not allowed with European options, an

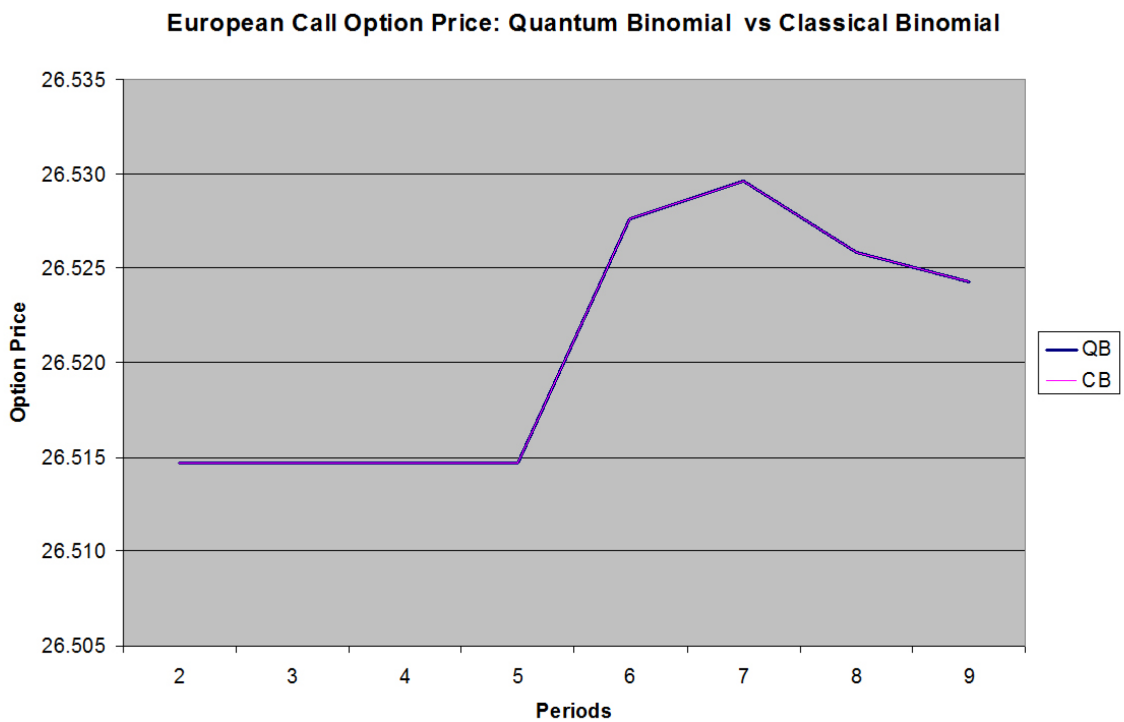


Figure 3.2: Comparison of the prices calculated by the classical binomial model and the quantum binomial model for a European call option. Note that they are identical.

American put option's price must be different. The quantum binomial model in its current format, is not well suited to price options where early exercise is possible. The reason is, that just like in the classical model, no analytical formula has been discovered to price American put options. A common classical approach has been to tackle this problem with binomial trees. Binomial trees can be created with the quantum model as well, but then it becomes exactly equivalent to the classical model and the possible advantage of using a quantum computer appears lost. The reason the advantage appears lost is, that every node needs to be evaluated individually to determine if early exercise is more desirable than continuing to hold the option

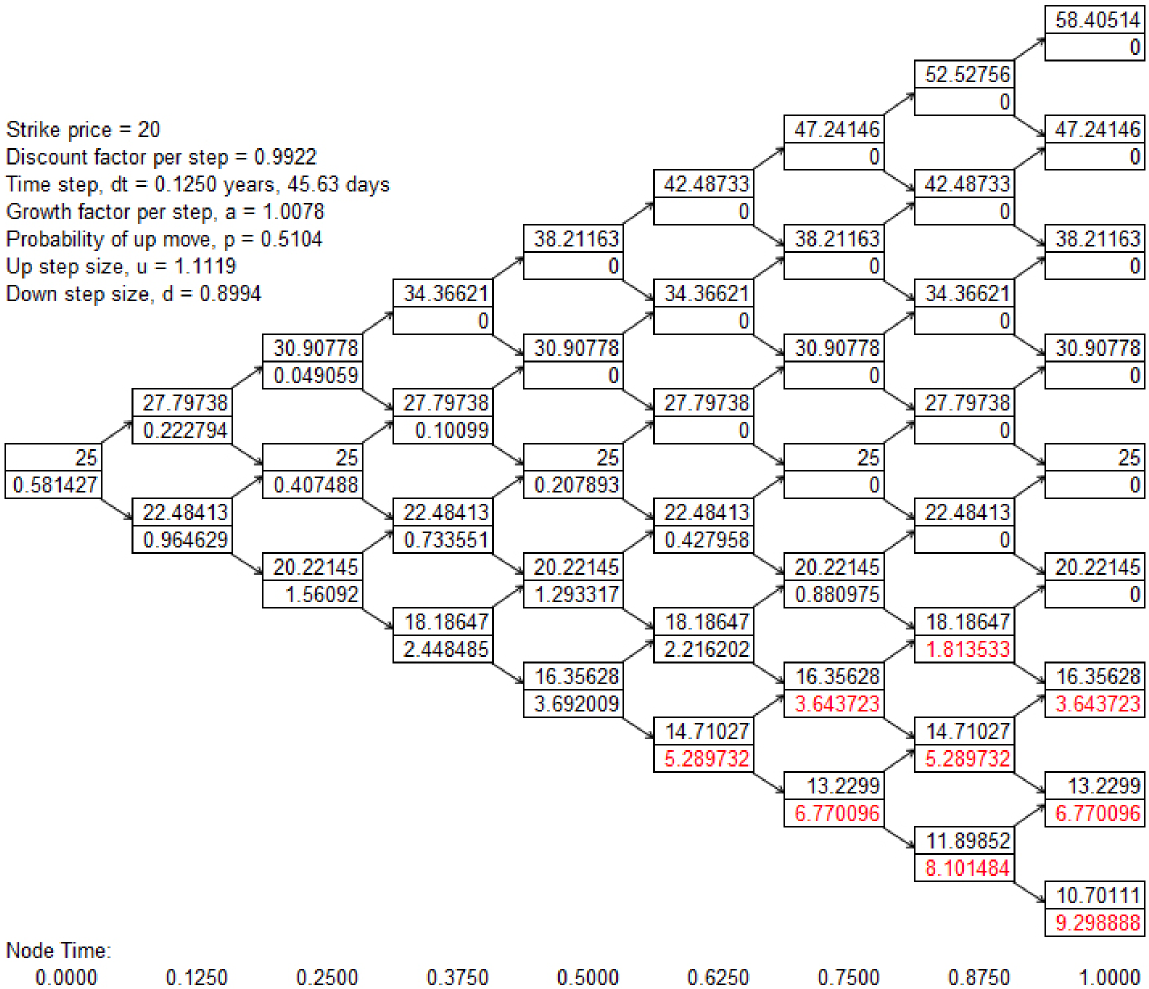


Figure 3.3: Eight period American put option classical binomial tree. The upper value is the stock price S_n and the lower value is the option price C_n. The values in red are situations where early exercise is desirable. $S_0 = 25$ and $K = 20$.

as shown in Figure 3.3. Further, one of the advantages of quantum computing is its ability to intrinsically deal with matrices efficiently and moving to a tree based algorithm that deals with individual nodes doesn't leverage this ability. That said, there is no reason to think that a quantum algorithm that does take advantage of quantum computing cannot be found for American put options.

Bermudan Options Because a Bermudan option can be exercised early like an American option, the quantum binomial model has the same issues as mentioned above for American options.

Barrier Options The quantum binomial model was extended to four styles of European Barrier options: Up-and-out, Down-and-out, Up-and-in and Down-and-in. The classical formulas that were extended are as a result of Chao et al. [33]. The quantum binomial model was not extended to the American version of these Barrier option styles for same reasons mentioned above for American options. When they are effective, the payoff of European Barrier call options take the form $[S-K]^+$ whereas European Barrier put options take the form $[K-S]^+$ just like vanilla European Options. The quantum binomial equation for the price of these European Barrier option styles, assuming the Maxwell-Boltzmann classical statistical distribution, are as follows; where $I_{\{\dots\}}$ denotes the indicator function[4] and S_B is the barrier level. For Up-and-out call options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[S_N - K]^+ I_{\{S_j < S_B\}}] \tag{3.20}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [S_0(1+b)^n (1+a)^{N-n} - K]^+ I_{\{S_n < S_B\}} \tag{3.21}$$

[4] $I_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$

For Up-and-out put options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[K - S_N]^+ I_{\{S_j < S_B\}}] \tag{3.22}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [K - S_0(1+b)^n (1+a)^{N-n}]^+ I_{\{S_n < S_B\}} \tag{3.23}$$

For Down-and-out call options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[S_N - K]^+ I_{\{S_j > S_B\}}] \tag{3.24}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [S_0(1+b)^n (1+a)^{N-n} - K]^+ I_{\{S_n > S_B\}} \tag{3.25}$$

For Down-and-out put options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[K - S_N]^+ I_{\{S_j > S_B\}}] \tag{3.26}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [K - S_0(1+b)^n (1+a)^{N-n}]^+ I_{\{S_n > S_B\}} \tag{3.27}$$

For Up-and-in call options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[S_N - K]^+ [1 - I_{\{S_j < S_B\}}]] \tag{3.28}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [S_0(1+b)^n(1+a)^{N-n} - K]^+ [1 - I_{\{S_n < S_B\}}] \tag{3.29}$$

For Up-and-in put options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[K - S_N]^+ [1 - I_{\{S_j < S_B\}}]] \tag{3.30}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [K - S_0(1+b)^n(1+a)^{N-n}]^+ [1 - I_{\{S_n < S_B\}}] \tag{3.31}$$

For Down-and-in call options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[S_N - K]^+ [1 - I_{\{S_j > S_B\}}]] \tag{3.32}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [S_0(1+b)^n(1+a)^{N-n} - K]^+ [1 - I_{\{S_n > S_B\}}] \tag{3.33}$$

For Down-and-in put options:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N} \rho_j)[K - S_N]^+ [1 - I_{\{S_j > S_B\}}]] \tag{3.34}$$

$$C_0^N = (1+r)^{-N} \sum_{n=0}^{N} \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} [K - S_0(1+b)^n(1+a)^{N-n}]^+ [1 - I_{\{S_n > S_B\}}] \tag{3.35}$$

The difference between simulating vanilla European options and European Barrier options is that the quantum binomial algorithm needs to determine when a Barrier event has occurred and make the option either effective or expire it worthless. To accomplish this, an enhancement to the step that calculates the payoff formula for the option style was developed to check for a barrier event and it was embedded into the $\mathcal{L}$ gate in the quantum circuit shown above.

When pricing Barrier options with the classical binomial model there are two types of errors that occur: *quantization error* and *specification error*. Quantization error tends towards zero as the number of periods approaches infinity, and effectively measures the differences between the current value of the binomial simulation and the analytical value. Specification error on the other hand comes from the fact that the barrier does not always align with the nodes on the binomial tree. As the number of periods simulated increases, the coarseness of the binomial tree reduces, which in turn reduces the impact of the specification error. There are various techniques that can be used to reduce specification error [13] including the use of trinomial trees as shown in Figure 3.4. With the quantum binomial model there is an additional source of error, which we will call *eigenvalue error*. Eigenvalue error results from the fact that the quantum binomial model is really operating at the leaf node level of the classical binomial model (as discussed in the previous sections) and that the leaf nodes are equal to the eigenvalues of the quantum operators. As a result of the coarseness, additional nodes are counted as part of the option value as can be seen in Figure 3.5. The red node is counted by the quantum binomial method but not in the classical binomial tree method as shown in Figure 3.6. Note that the same techniques used to reduce specification error can be used to reduce eigenvalue error.

The results of simulating a down-and-out Barrier European call option are shown

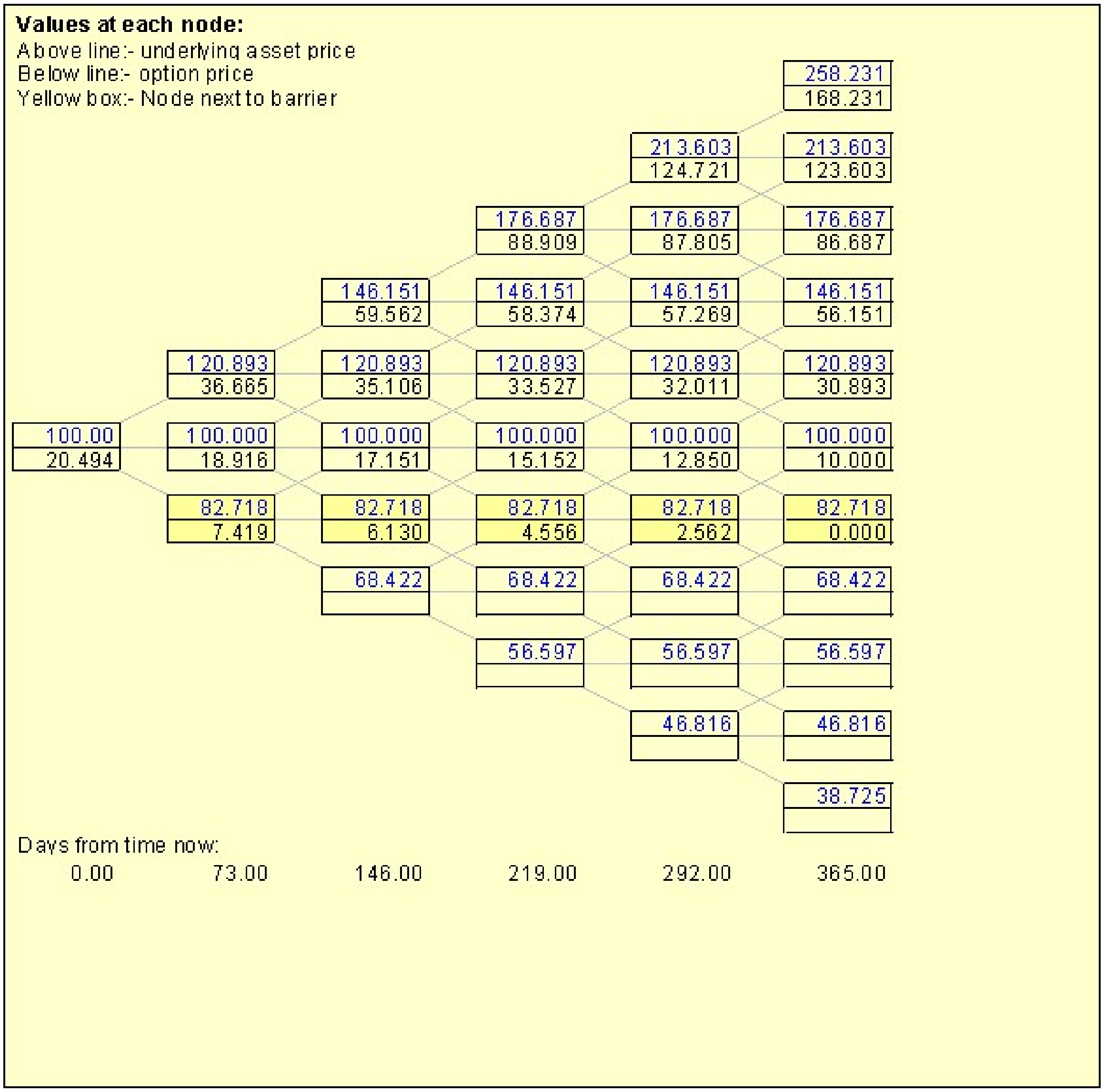


Figure 3.4: Five period down-and-out European Barrier call option classical trinomial tree. $S_0 = 100$, $K = 100$ and $B = 120$.

in Figure 3.7 for the quantum binomial method, the classical binomial method and the classical trinomial method. In general the quantum binomial method will produce a higher option price than the classical models because of the additional nodes that will be counted as part of the eigenvalue error mentioned above.

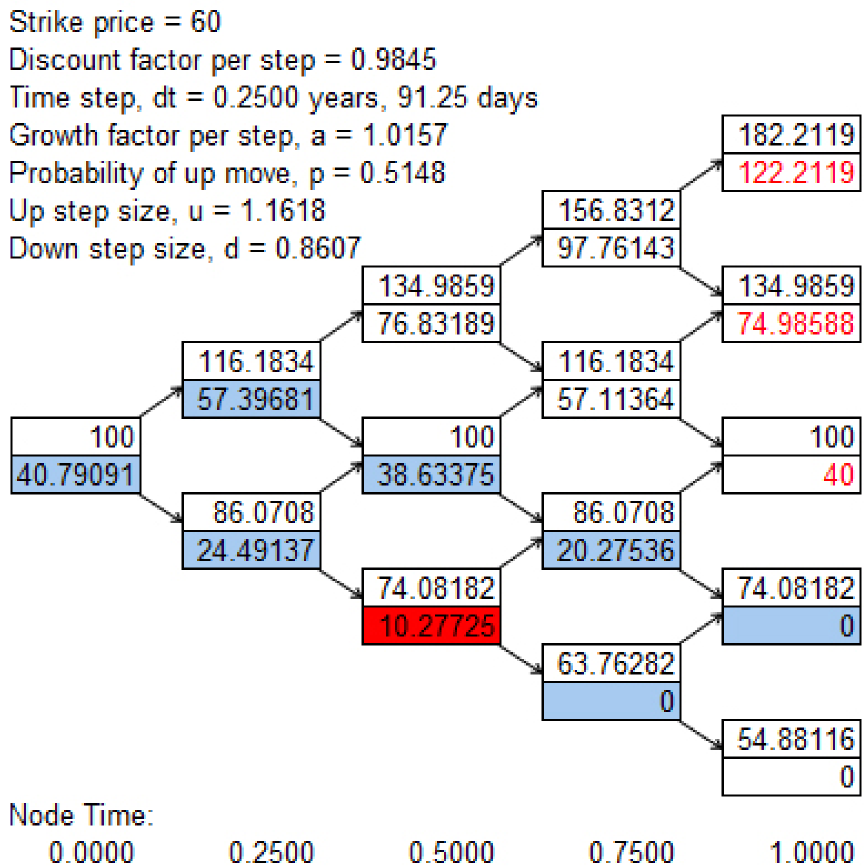


Figure 3.5: Four period down-and-out European Barrier call option effective classical binomial tree for the quantum binomial model. $S_0 = 100$, $K = 60$ and $B = 80$.

Asian Options As discussed earlier, an Asian option is also known as an average option. Asian options are based on either the geometric average or the arithmetic average of the underlying security over a specified time period. Because the average of the underlying security depends on the values the security took during the simulation, it is often referred to as a path dependent option. As will be shown below, path

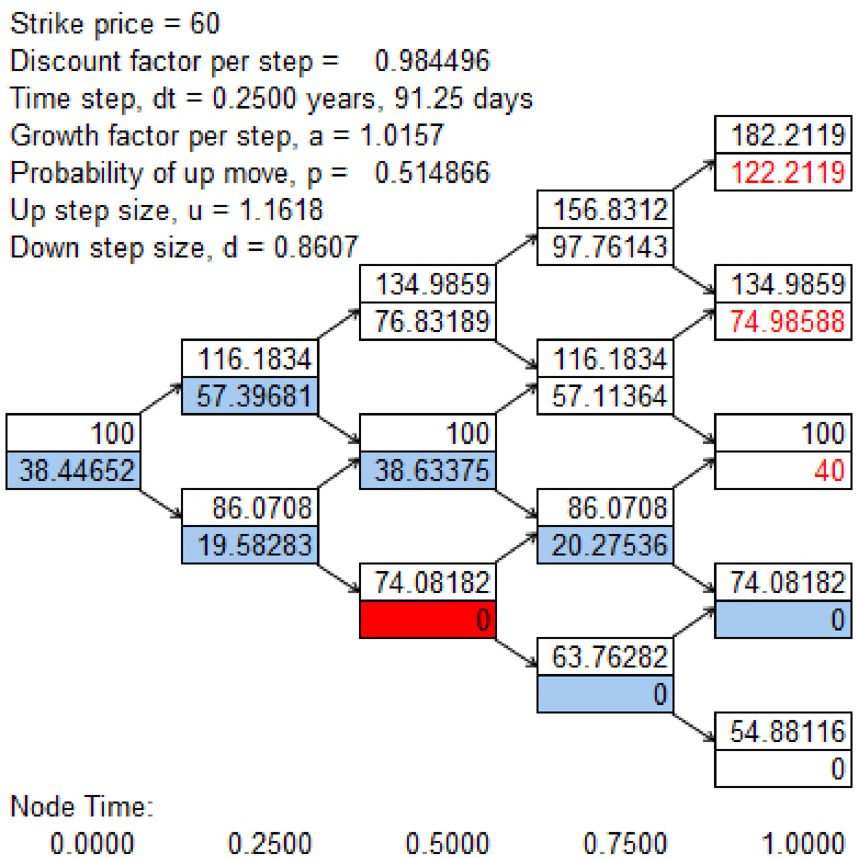


Figure 3.6: Four period down-and-out European Barrier call option classical binomial tree. $S_0 = 100$, $K = 60$ and $B = 80$.

dependence poses a problem for the quantum binomial model. Just as was discussed with American options above, Asian options also require that each node of the tree be evaluated. However, instead of considering early exercise only, the average value a security took while traversing each path in the tree needs to be calculated. Again, a quantum binomial tree could be created to overcome this problem but as mentioned

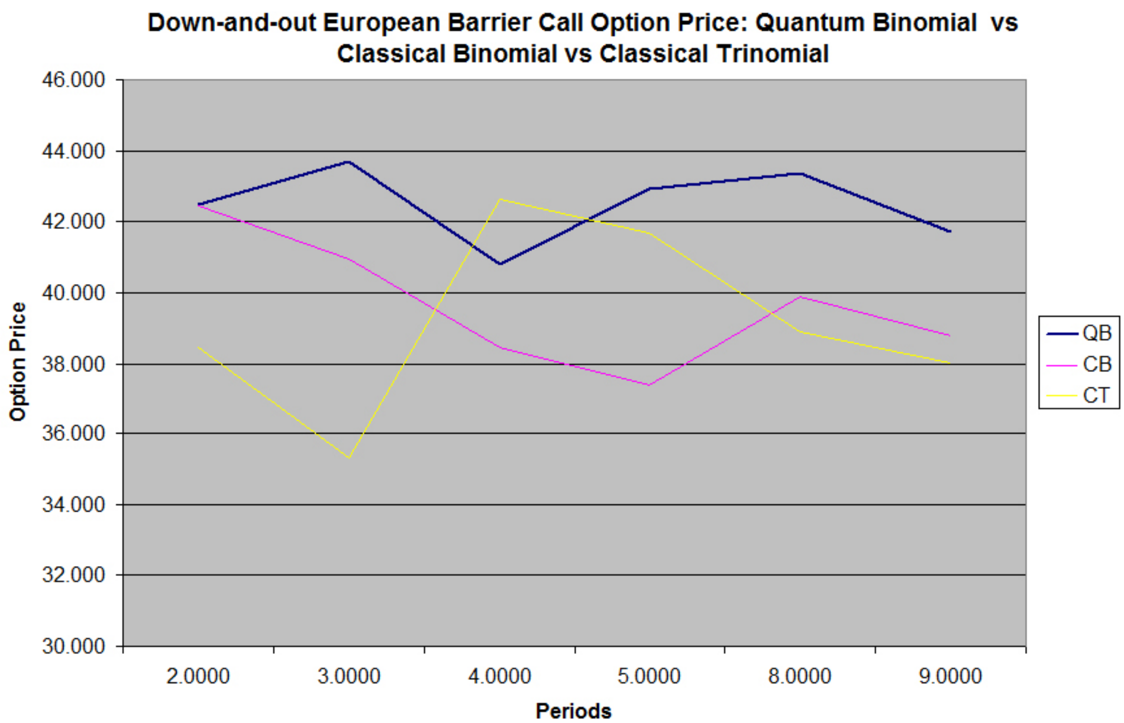


Figure 3.7: Comparison of the quantum binomial model, the classical binomial model and the classical trinomial model for the price of a down-and-out Barrier European call option. $S_0 = 100$, $K = 60$ and $B = 80$.

above, it becomes exactly equivalent to the classical model, and the possible advantage of using a quantum computer appears lost, as the quantum computers intrinsic ability to operate with matrices is not being leveraged. As an illustration, consider the following, the payoff of an Asian option is defined as follows at maturity $t + T$:

$$[S_{AVG}(t + T) - K]^+ \tag{3.36}$$

where $S_{AVG}(t + T)$ is the arithmetic average of the underlying asset prices attained during the lifetime of the option T and where K is the strike price. The detail of the arithmetic average is as follows:

$$S_{AVG}(t+T) = \frac{1}{n+1}\sum_{i=0}^{n} S(t+i\Delta t) \tag{3.37}$$

Replacing the payoff formula in the quantum multi-step binomial model used for European options results in the following price for an Asian option:

$$C_0^N = \mathrm{tr}[(\bigotimes_{j=1}^{N}\rho_j)(S_{AVG}-K)^+] \tag{3.38}$$

$$= \mathrm{tr}[(\bigotimes_{j=1}^{N}\rho_j)(\frac{1}{j+1}\sum_{i=0}^{j} S(t+i\Delta t-K)^+]) \tag{3.39}$$

While it appears simple to extend the Asian option by simply substituting S_N for S_{AVG}, it is not. The reason it is not is; the calculation of S_{AVG} requires considering of all of the 2^N paths the stock's value can take. How to accomplish this, while being constrained by the finite dimensional Hilbert space of quantum computing, is not clear. As mentioned in the related works section, Baaquie's research and others like Matacz [25] have used Feynman's path integrals to attempt to price Asian options. Essentially, they reformulate classical results into quantum field theory language to obtain equivalent results. This does not directly translate into a way to implement their approaches with a quantum computer. Further it remains an open question as to whether or not quantum field theory can be simulated efficiently on a quantum computer at all. See a discussion on this topic by Ahrensmeier [3].

In this section, only Asian options based on the arithmetic average of the underlying security were considered, although the same ideas can be extended to Asian options based on a geometric average.

3.3.1 Risk Free Unit Ball and Sigma

Another interesting result is, as Chen claimed, by varying the parameters that make up A and ρ within the unit ball of $\mathbf{R}^3$ the value of the quantum European call option price equals the risk-free price. In addition, just as in the classical one step model, when σ changes, the option price does not.

Chapter 4

Conclusion

There were several outcomes from this research. The first is that, as Chen proposed, the simulations show that when Maxwell-Boltzmann statistics are assumed, the quantum binomial model produces option prices that are equivalent to the classical binomial model. Second is that European and Barrier options can be priced efficiently on a quantum computer after the circuit has been built. The time complexity of creating and running the circuit for the first simulation is $O(2^{(N-\tau)} + (N-\tau)\log_2(N-\tau))$ where τ is the boundary where the option becomes in-the-money. If the same circuit is used to run another simulation, but the volatility σ and the number of periods N are kept the same, the time complexity is $O((N-\tau)\log_2(N-\tau))$. Analysis shows that this quantum algorithm is in the **BQP** quantum computational complexity class. That said, the quantum algorithm implemented does not take into account early exercise, however, it does preserve all of the information that would be contained within a full binomial tree. So although it runs slower than a classical binomial algorithm that does not take into account early exercise, it does run faster than one that does.

The algorithm can also be useful as a subcomponent to a larger quantum algorithm where preserving the quantum state of the system is required. Further, because the matrix representing every period of the simulation is generated, it seems reasonable to expect that there should be a way to incorporate early exercise into the algorithm without much change in its complexity. If this is possible, then option styles where early exercise needs to considered, could be priced exponentially faster, as classical binomial algorithms for these option styles have a space and time complexity of $O(N^2)$. The key reason for this possible efficiency is, that these option styles can be priced using quantum operators and density matrices, which are well suited for quantum computers. As a result, each individual binomial state may not need to be considered and the processing could be done on aggregates. Modifying the algorithm to consider early exercise has been left to future research. The final outcome of this research is that Asian options require a different approach. The reason is, that not only does each individual binomial state of an Asian option need to be considered, but there is also an exponential number of paths to consider. As a result, pricing Asian options with a classical binomial tree has a space and time complexity of $O(2^N)$. Further, a method to price these options efficiently with a quantum algorithm is not clear. Although the quantum model can also be represented using a binomial tree, it basically becomes equivalent to the classical model, and the possible advantage of using a quantum computer appears lost. One approach to this problem, is to look at efficiently simulating binomial trees on a quantum computer. Another approach, is to look at leveraging research for Asian options that use quantum field theory frameworks, and find a way to efficiently simulate quantum field theory on a quantum computer.

Bibliography

[1] Quantum circuit implementing Shor's algorithm. The Center for Bits and Atoms, 2002. http://cba.mit.edu/docs/02.06.NSF/images/circ.jpg.

[2] L. Accardi and A. Boukas. The quantum Black-Scholes equation. *GJPAM*, 2(2):155–170, 2006.

[3] D. Ahrensmeier. Can a quantum computer simulate quantum field theory - efficiently? *Canadian Journal of Physics*, 86:617–621, 2008.

[4] B. E. Baaquie. *Quantum Finance: Path Integrals and Hamiltonians for Options and Interest Rates*. Cambridge University Press, 2007.

[5] B. E. Baaquie, C. Coriano, and M. Srikant. Quantum Mechanics, Path Integrals and Option Pricing: Reducing the Complexity of Finance. *ArXiv Condensed Matter e-prints*, Aug. 2002.

[6] J. Bernoulli. *Jacobi Bernoulli ars conjectandi : accedit tractatus de seriebus infinitis, et epistola gallice scripta de ludo pilae reticularis*. Impensis Thurnisiorum Fraetrum, Basileae, 1713.

[7] F. Black and M. Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81(3):637–654, 1973.

[8] B. Boghosian. Simulating quantum mechanics on a quantum computer. *Physica D*, 120(1-2):0–42, 1998.

[9] G. Brassard. Teleportation as a quantum computation. *Physica D*, 120(1-2):43–47, 1998.

[10] Z. Chen. Quantum theory for the binomial model in finance theory. *Journal of Systems Science and Complexity*, 17(4), 2004.

[11] I. J. Cox, J. and S. Ross. A theory of the term structure of interest rates. *Physica A*, 51:385–407, 1985.

[12] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, September 1979.

[13] E. Derman, I. Kani, D. Ergener, and I. Bardhan. Enhanced numerical methods for options with barriers. *Financial Analysts Journal*, 51(6):65–74, 1995.

[14] T. G. Draper. Addition on a quantum computer. 2000. http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/0008033.

[15] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6&7):467–488, 1982.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, 1996.

[17] T. F. Havel and C. J. L. Doran. A Bloch-sphere-type model for two qubits in the geometric algebra of a 6-d Euclidean vector space. In *Quantum Information and Computation II*, volume 5436, page 93, 2004.

[18] E. Haven. A Black-Scholes Schrodinger option price: 'bit' versus 'qubit'. *Physica A*, 324(1):201–206, 1 June 2003.

[19] E. Haven. A discussion on embedding the Black-Scholes option pricing model in a quantum physics setting. *Physica A*, 304(3-4):507–524, 15 February 2002.

[20] J. Hull. *Options, Futures, and Other Derivative Securities.* Prentice-Hall, Englewood-Cliff, N.J., sixth edition, 2006.

[21] J. Hull and A. White. The pricing of options on interest rate caps and floors using the Hull-White model. *Advanced Strategies in Financial Risk Management*, 4:59–67, 1990.

[22] S. P. Jordan. Quantum computation beyond the circuit model. 2008. http://www.citebase.org/abstract?id=oai:arXiv.org:0809.2307.

[23] A. Khrennikov. Classical and quantum randomness and the financial market. *ArXiv e-prints*, 704, Apr. 2007.

[24] Y.-D. Lyuu. *Financial engineering and computation: principles, mathematics, and algorithms.* Cambridge University Press, New York, NY, USA, 2002.

[25] A. Matacz. Path dependent option pricing: The path integral partial averaging method. *Journal of Computational Finance*, 6(2):79–104, 2002.

[26] R. C. Merton. Theory of Rational Option Pricing. *Bell Journal of Economics*, 4(1):141–183, 1973.

[27] K. Meyer. Binomial quantum finance simulation. University of Manitoba COMP 7570 Final Project, 2006.

[28] J. A. Miszczak. Initialisation of quantum registers based on probability distribution. Technical Report N206 013 31/2258, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Baltycka 5, 44-100 Gliwice, Poland, March 2007.

[29] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[30] R. K. Niven and M. Grendar. Generalized Maxwell-Boltzmann, Bose-Einstein, Fermi-Dirac and Acharya-Swamy Statistics and the Polya Urn Model. *ArXiv e-prints*, 808, Aug. 2008.

[31] E. W. Piotrowski, M. Schroeder, and A. Zambrzycka. Quantum extension of European option pricing based on the Ornstein Uhlenbeck process. *Physica A Statistical Mechanics and its Applications*, 368:176–182, Aug. 2006.

[32] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[33] C. Sun, J.-Y. Yang, and S.-H. Li. On barrier option pricing in binomial market

with transaction costs. *Applied Mathematics and Computation*, 189(2):1505–1516, 2007.

[34] G. E. Uhlenbeck and L. S. Ornstein. On the theory of Brownian motion. *The Journal of Political Economy*, 36:823–841, 1930.

[35] G. F. Viamontes, I. L. Markov, and J. P. Hayes. Improving gate-level simulation of quantum circuits. *Quantum Information Processing*, 2:347, 2003.

[36] J. Watrous. Quantum computational complexity. *ArXiv e-prints*, 804, Apr. 2008.

[37] World Federation of Exchanges. *Market Capitilization*, 2005. www.world-exchanges.org.