

Foreword

This report was originally intended as a report for the course Introduction to Quantum Computing at the University of Twente. Both the report and the course are provided by ING employees. Initially, the report was meant to show maturity in quantum computing of the students attending the course. However, it became quickly interesting to upstream the report into a tutorial for, mostly colleagues, who would like to have a step-by-step introduction to quantum computing and to execute Monte Carlo simulations on such computers. The report is going to be a living document and will be updated from time to time as the field of Monte Carlo simulations on quantum computers advances and the author's experiences too. Lastly, the Monte Carlo simulations are applied in the domain of (quantitative) finance, thus, covering the financial problems too in the report. We hope that this report meets its requirements as a standalone document for the quantum computing enthusiasts in the financial world.

1 Introduction

Banks and other financial institutes employ vast computational resources for pricing and risk management of financial assets and derivatives. Without the computational resources evaluating complex mathematical models today's financial markets would be in grave danger and could not operate efficiently or at the scale it does nowadays. Examples of assets on financial markets are stocks, bonds and commodities. The assets are a base for more complex financial instruments called derivatives. A financial derivative, as its name suggests, derives its price from the future price or a price trajectory of at least one asset. As the nature of risky assets is stochastic, determining a fair price of a derivative (contract) is a challenge. A mathematical model that can be used for this pricing model is called the Black-Scholes-Merton model. In Section ?? this model is discussed in more depth and its applications on derivatives is addressed.

The Black-Scholes-Merton model is only solved analytically for very simple derivatives. For example, a simple derivative is a derivative whose payoff only relies on the future price at maturity time. A more complex derivative would be one where the payoff depends on the trajectory towards some future price at maturity time. Unfortunately, these complex derivatives have path-dependence embedded in the payoff function and combined with the stochastic nature of the underlying asset(s), it becomes infeasible (or perhaps impossible) to find an analytical solution for the pricing problem. Due to the stochastic nature of the underlying risky asset(s), it is wise to revert to a method where the expectation value of the future price (or the trajectory) is calculated by repeatedly and randomly sampling from a probabilistic distribution describing the underlying asset price (development). This method, called the Monte Carlo method, has long been explored in physics, chemistry, biology, finance, meteorology and many other domains where analytical solutions are infeasible and stochastic behaviors are observed. Random sampling from a distribution yields a higher accuracy (i.e., lower error) as the number of samples is growing. The downside of increasing the number of samples is the increased cost of computational power. A ten-fold improvement in accuracy suggests a hundred-fold longer runtime. Monte Carlo simulations can become rapidly intractable due to the scaling illustrated in the previous sentence. Thus, perhaps by changing the nature of deterministic processing units in computers to stochastic processing units, such as quantum processing units, a better scaling might be attained. Therefore, this report explores the opportunities and challenges

of quantum computers regarding Monte Carlo simulations applied in the financial industry. For a more in-depth introduction on Monte Carlo simulations, please consult Section ??.

A quantum algorithm and computer capable of performing Monte Carlo simulations of financial derivatives (and the pricing problems) in a consistent and reliable way would be of utmost importance. This document discusses and shows which algorithms on quantum computers might be helpful in risk analytics and derivatives pricing.

1.1 Introduction to Quantum Computing

Gated quantum computing is similar to classical computing with logical circuits. A logical circuit is composed of bits and gates. A bit is a binary digit carrying one of the following two values exclusively: 0 (*false*) or 1 (*true*). A quantum bit, abbreviated as *qubit*, is a quantum state with two base states $|0\rangle$ and $|1\rangle$ and does not exclude the presence of both states simultaneously (which is called the *superposition* principle). How and how much a base state is contributing to the qubit is determined by the amplitude (α , and β in the equation below) of each base state. Upon measuring a quantum state, hence forcing it into a classical realm, the superposition principle collapses and only one single deterministic state must be acquired. The resulting state is retrieved by a stochastic process in which the probability distribution of the base states is characterized by the absolute value squared of amplitudes. As the sum of all probability coefficients of a probability distribution must be one, the sum of all absolute value squared of amplitudes must be one too. A qubit is then defined as the equation below following both an algebraic and a vector representation respectively:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \text{ with } (\alpha, \beta) \in \mathcal{C}, \text{ and } |\alpha|^2 + |\beta|^2 = 1 \quad (1)$$

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2)$$

Logical gates take in a Boolean input and return a Boolean output as followed from a Boolean function embedded in that gate. For example, a NOT-gate has a Boolean formula embedded in it that inverts the binary value of an input bit. To illustrate, an input bit with value 0 would become an output bit with value 1. A quantum gate follows the same principle. The quantum gate has an input quantum state (or an amplitude vector), follows a linear algebraic formula (or a matrix) and returns an output quantum state (again, an amplitude vector). One notable difference between classical and quantum computing is the number of bits and qubits put in and returned. In classical computing the number of input and output bits may differ, for example, the AND-gate, would require two input bits and only returns a single output bit. In quantum computing a quantum gate performs an operation on (a set of) qubits and solely transforms the states of these qubits only. More technically, a quantum gate is a unitary matrix. The unitary property, i.e., the complex conjugate transpose of a matrix multiplied by the original matrix is an identity matrix, ensures that the norm of the quantum state vector is preserved (and remains one). All unitary matrices are square too. The property of being square results in the fact that the number of qubits is preserved during the operation. An example of a quantum gate deployed on a single qubit is given below where the amplitudes of the base states are swapped:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

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

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

Qubits are technically quantum states, hence, are governed by a wave function as followed from/by Schrödinger's equation. The wave nature allows for *interference* where amplitudes of a base state may constructively or destructively interfere. Below is an example of interference upon using an Hadamard gate on a generic qubit.

$$\begin{aligned} H|\psi\rangle &= \sqrt{\frac{1}{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \\ &= \sqrt{\frac{1}{2}} \begin{bmatrix} \alpha + \beta \\ \alpha - \beta \end{bmatrix} = \\ &= \sqrt{\frac{1}{2}}(\alpha + \beta)|0\rangle + \sqrt{\frac{1}{2}}(\alpha - \beta)|1\rangle \end{aligned}$$

If $\alpha = \beta$, then the $|1\rangle$ -state would be diminished completely. Even though the qubit started in a superposition of two base states, due to interference, the resulting qubit is now characterized by a single base state only. This does not mean the other base state is removed entirely, it is solely absent (until some other operation would redistribute the amplitudes such to acquire a non-zero amplitude on both base states).