Monte Carlo simulation techniques

-The development of a general framework

Master's Thesis carried out at the
Department of Management and Engineering,
Linköping Institute of Technology
and at Algorithmica Resarch AB

by

Emma Nilsson

LIU-IEI-TEK-A--09/00572--SE

Supervisors

Peter Hultman (IEI)
Peter Alaton & Niclas Holm(company)

Acknowledgement

The work presented in this report has been performed the fall 2008 (from August to December) and it is a Master's thesis carried out in collaboration with Algorithmica Research AB.

I would like to thank my supervisor at Linköping University; Peter Hultman for all the help and the support during the work, and especially for all the discussions about my project. I would also like to thank my supervisors at Algorithmica Research AB; Niclas Holm and Peter Alaton for helping me through the work. Special thanks should also be given to the employees at Algorithmica Research AB, because when I needed help they always took the time to answer my questions.

Finally I would like to give my appreciation to my friends and family, because of their great encouragement.

Stockholm January 2009
Emma Nilsson

Executive Summary

Algorithmica Research AB develops software application for the financial markets. One of their products is Quantlab that is a tool for quantitative analyses. An effective method to value several financial instruments is Monte Carlo simulation. Since it is a common method Algorithmica is interesting in investigating if it is possible to create a Monte Carlo framework.

A requirement from Algorithmica is that the framework is general and this is the main problem to solve. It is difficult to generate a generalized framework because financial derivatives have very different appearances. To simplify the framework the thesis will be delimitated to European style derivatives where the underlying asset is following a Geometric Brownian Motion.

The definition of the problem and delimitation were defined gradually, in parallel with the review of literature, this to be able to decide what purpose, and delimitations that is reasonable to treat. Standard Monte Carlo requires a large number of trials and is therefore slow. To speed up the process there exist different variance reduction techniques and also Quasi Monte Carlo simulation, where deterministic numbers (low discrepancy sequences) is used instead of random. The thesis investigated the variance reduction techniques; control variate technique, antithetic variate technique, and the low discrepancy sequences; Sobol, Faure and Halton.

Three test instruments were chosen to test the framework, an Asian option and a Barrier option where the purpose is to conclude which Monte Carle method that performs best, and also a structured product; Smart Start, that is more complex and the purpose is to test that the framework can handle it.

To increase the understanding of the theory the Halton, Faure and Sobol sequence were implemented in Quantlab in parallel with the review of literature. The Halton and Faure sequences also seemed to perform worse than Sobol so they were not further analyzed.

The developing of the framework was an iterative process. The chosen solution is to design a general framework by using five function pointers; the path generator, the payoff function, the stop criterion function and the volatility and interest rates. The user specifies these functions by him/her given some obligatory input and output values. It is not a problem-free solution to use function pointers and several conflicts and issues are defined, therefore it is not recommended to implement the framework as it is designed today.

In parallel with the developing of the framework several experiments on the Asian and Barrier options were performed with varying result and it is not possible to draw a conclusion on which method that is best. Often Sobol seems to converge better and fluctuates less than standard Monte Carlo. The literature indicates that it is important that the user has an understanding of the instrument that should be valued, the stochastic process it follows and the advantages and disadvantages of different Monte Carlo methods. It is recommended to evaluate the different method with experiments, before deciding which method to use when valuing a new derivative.

Table of Contests

1   Introduction

This chapter is the opening chapter of the Master's thesis and it describes the background of the thesis and the problems are defined. The thesis then presents the purpose and the research questions. Finally the delimitations are formulated.

1.1   Background

This Master's thesis project is carried out in collaboration with Algorithmica Research AB who develops software application for the financial markets. One of their products is Quantlab which is a tool for analysts and traders. Quantlab can be used for financial calculations based on time-series and real-time data, and is available in two versions; the Developer edition and the User edition. The Developer edition allows the user to build quantitative models and views, while the User edition only gives access to the graphical interface and gives the user possibilities to change different parameters like dates, markets and instruments.

Quantlab contains an expression language called QLang, which is a high-level object-oriented programming language, allowing the user to create new functions. With QLang follows an extensive library of financial functions and classes, which facilities the programming. It is also easy for the user to add new library files. The files and classes for Quantlab can also be written in C++ and all functions are available from Microsoft Excel, Visual Basics and .NET.

Today there exist several kinds of financial derivatives, where standard instruments often have an analytic solution while more complex instrument often must be valued numerically. When an instrument depends on several state variables, like mortgage backed securities or path-dependent derivatives an effective method to value the instrument is Monte Carlo (MC) simulation. Since MC simulation is a very useful method Algorithmica is interested in developing a MC framework. Given that the customers have different kind of derivatives the MC should be general.

1.2   Definition of the problem

The main problem of the Master's Thesis will be how to generate a general MC framework where the user can specify the derivative and choose what kind of MC method to use. The Master's thesis will examine whether it is better to let the users write a large part of the code by themselves or if is it better to specify the different cases.

In MC simulation the derivative is written as a multidimensional integral that may be approximated with a sum by using random numbers. A disadvantage with standard MC simulation is that the method requires a large number of trials and is therefore slow. To speed up the process there exist different variance reduction techniques and also Quasi Monte Carlo (QMC) simulation, where the integral is evaluated by using deterministic numbers (low discrepancy sequences) instead of random. (Boyle & Tan, pp. 2-3).

The literature is contradictory concerning which method is best, standard MC or QMC. According to numerical research, low discrepancy sequences are more effective in modest dimensions, that is $d \leq 30$, whereas MC is better with high dimensions as long as the number of trials not is too large. Concerning financial instruments these rules do not apply. Paskov (pp. 2-3) shows that when pricing a Collaterized Mortgage Obligation in 360 dimensions, the low-discrepancy sequence Sobol is far better than standard MC, even if variance reduction techniques are used. Other studies gives other results, hence it is not possible to conclude when to use which method. For this reason several methods will be evaluated and if they provide a good result it is desirable that they will be implemented in the framework.

On a more specific level the Master's thesis will examine which MC method that is convenient to use on an Asian and a Barrier option. Regarding the variance reduction techniques,

it will be studied whether they are appropriate to use and if they can be used together with the low discrepancy sequences.

The financial properties on the instruments also arises some problem concerning the random numbers. If the instruments are assumed to follow a Geometric Brownian Motion, the random number must be normally distributed. When valuing a portfolio, the assets are often correlated with each other; which also means that the random number must be correlated.

1.3 Purpose

The purpose of the Master's thesis is to examine the best way to develop and implement a general MC framework. The framework should contain standard MC, MC with variance reduction techniques and also QMC. Finally the thesis shall investigate the performance of the different methods.

1.4 Delimitations

Even though the purpose is to develop a general framework the Master's thesis will delimit the financial instrument to European style derivatives where the underlying asset is following a Geometric Brownian Motion. Chosen test instrument are:

- A structured product called Smart Start distributed by Kaupthing Bank Sverige, for description of the product see Appendix I.
- Asian option
- Barrier option

Concerning the variance reduction techniques the framework will treat:

- Control Variate technique
- Antithetic Variate technique

The chosen low-discrepancy sequences are:

- Halton
- Faure
- Sobol

The general MC framework will not deal with the problem of estimating parameters like volatility rate of return etcetera, the parameters will be seen as input parameters and it is up to the user to define them.

2 Methods of research

This chapter describes the method that is used in the study, first the data types are presented and thereafter the mode of procedure. Furthermore the quality of the chosen method is discussed and analyzed.

2.1 The collection of data

The collection of data can be either qualitative or a quantitative. (Lekvall & Wahlbin, p. 200) In this thesis the most of the data is collected from several different experiments, which is a more quantitative research. The data can also be either primary data or secondary data; primary data is data that is collected by the researcher while secondary data is data that already exists and is already published. This Master's thesis will use both; the primary data will be collected from the experiments, while the secondary data will be taken from old experiments result that is done by other researches.

2.2 The procedure

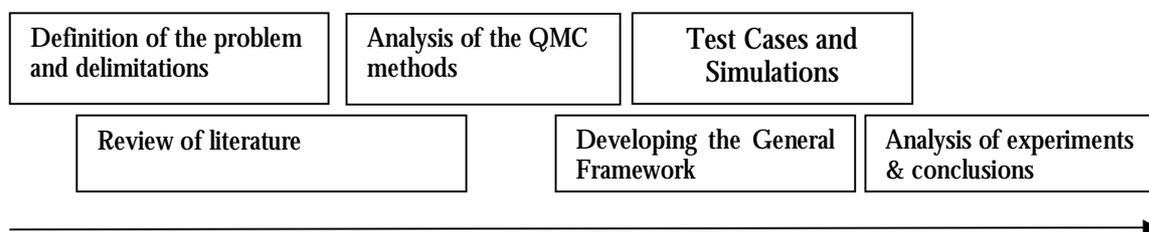Figure 1 presents the procedure that will be used during the Master's thesis.

| Definition of the problem and delimitations | Analysis of the QMC methods | Test Cases and Simulations |
| Review of literature | Developing the General Framework | Analysis of experiments & conclusions |

Figure 1. The mode of procedure

2.2.1 Definition of the problem and delimitations

The problem will be defined gradually in parallel with the review of literature, this to be able to decide what purpose, questions and delimitations that is reasonable to treat in a Master's thesis.

The framework will be delimited to European style derivative because the American derivatives are difficult to value with MC simulation. It will also be delimitated to an instrument where the underlying value is following a Geometric Brownian Motion as a stochastic process can be very different and therefore it is difficult to generalize. The Geometric Brownian Motion is chosen because the underlying asset of a derivative often follows this process.

The test instruments are chosen carefully with different purposes; the Asian option will be simulated with geometric average since it has an analytic value to compare with, and here the main purpose is to conclude which MC method that performs best. The Barrier options, has the same purpose but with a barrier implemented, and both an Up-and-out option and a Down-and-out option will be simulated to see if there are some difference. The purpose is also to see whether it is possible to conclude that a method always is the best.

For the Smart Start the purpose is not to price the instrument correctly, this because the parameters like volatility, dividend, correlation, previous price etcetera must be estimated from data and this does not lie within the Master's thesis. The main purpose with the Smart Start is to check that the framework is general and can handle an instrument with the properties that Smart Start has for example a specific pay off function and three correlated assets etcetera.

2.2.2 Review of literature

A major part of this Master's thesis will be the review of literature. The main reference to use is the book "*Monte Carlo Methods in Financial Engineering*" by Glasserman, but also several scientific research and different articles.

2.2.3 Analysis and Experiments

In parallel with the review of literature, the Sobol, Halton and Faure sequence will be implemented in Quantlab, much to increase the understanding of the theory. Experiments that evaluate the performance of the three QMC methods will also be done.

The developing of the general framework and the experiments of the test cases will be an iterative process, where the framework will be updated several of times when implementing and exploring new solutions. Before setting up the test cases some experiments must be done to assure that the test cases cover a big area of possible outcomes. The programming language is QLang.

2.3 Method Quality

All methods can be questioned, and the chosen method for this Master's thesis is analyzed on the parameters validity, reliability and possible sources of errors.

2.3.1 Validity

The validity is whether the used method investigates and measures what was supposed to be measured (MCNeill & Chapman, p. 9). The main purpose of this Master's thesis is to develop a general framework for Monte Carlo simulations.

From Algorithmica it is desirable that the framework can handle different stochastic processes. Of this reason the framework will be developed so the user may write the process himself with some obligatory input values, but the thesis cannot guarantee that these input values are enough for generating all kinds of paths, for this further analysis are required. Even within the delimitations to European style derivatives, the thesis cannot guarantee that very complex derivatives might be simulated in the framework.

The thesis investigates a proportionally small amount of test cases compared with other researches. This might imply that some of the conclusion that may be drawn will be drawn on an appearance that is more a random than a constant behaviour. To avoid this several simulations will be done where for example the seed is changed and if an instrument shows a very divergent instrument it will be investigated further and more test cases can be set up.

2.3.2 Reliability

Reliability means that if anybody else would use the same methods they will come up with the same results (MCNeill & Chapman, p. 9). If someone is doing exactly the same experiment, with the chosen test cases, the same seed on the random number etcetera the simulations results will be the same. It is not sure that the time will be the same, since this parameter also depends on the computer performance.

Concerning the general framework it is also likely to believe that the chosen implementation techniques will look different, if they were done by another person. The writer of the thesis did not have any experience from the programming language QLang before starting with the work. It is possible that a more experienced programmer would come up with a more effective solution.

2.3.3   Source of error

Concerning the review of literature, the reliability can be questioned in several articles, since they are written by unfamiliar postgraduate students that refer to other scientists. Instead of using the articles the origin source will be used to confirm the statement. If it is not possible to find the origin source the statement will be confirmed with other articles and researches that say the same thing and refer to the same source. But there can still be resources that are not correct since the personal experience and beliefs affects the researchers.

The experiments in the thesis are performed on some chosen test cases. This is a potential source of error, because the conclusions are based on the experiments and they could have been different if further experiments with other test instruments had been executed.

Another potential source of error is the code and calculations. Even if the code is debugged, compiled and the result has been compared with the analytic value, it can still be errors, that is giving a wrong result. The same is about the formulas and calculations. Everything will be done by one single person, and it can be difficult to correct and find one's one fault.

The thesis is written in English, because most of the literature about the subject is written in English. But since the mother tongue of the writer is Swedish it is possible that the literature sometimes had been misunderstood.

3 Fundamental financial theories

This chapter starts with a description of the structure of derivatives and the principles behind valuing them. These are important theories that the reader needs to comprehend for understanding the Master's thesis. If the reader already has a good financial knowledge, it is possible to pass over the chapter and continue with the theoretical framework.

3.1 Derivatives

On the financial market derivatives are important financial instruments. Hull (p. 1) defines a derivative as "*A financial instrument whose value depends on (or derives from) the values of other, more basic, underlying assets.*". There exist a large number of different derivatives, common examples are equity and debt derivatives, where the payoff depends on a stock or a bond price, but there are also derivatives depending on almost any variables, like weather or electricity etcetera.

A derivative is a contract between two parties, one taking a long position in the derivative (buys the assets) another taking a short position (sells the assets). The contract also defines the parties' rights and obligations towards each other.

3.1.1 Options

There are two standard types of options; put and call options. A put option gives the holder the right to sell an asset for a predetermined price at a certain date, while the call option gives the holder the right to buy it. The options can be American or European, where an American style allows the option to be exercised at any time up to the maturity date, while a European option only allows exercising at maturity date.

The payoffs for a long position for European options are:

$$\text{Put option: } \max\{0, K - S_T\}$$
$$\text{Call option: } \max\{0, S_T - K\}$$

where K is the strike price and S_T is the final price of the underlying assets. The underlying assets are often stocks, but currencies, stock indices and futures are also very common (Hull J. C., pp. 181-185).

During the end of 1970's the trading with standard options exploded and several financial institution searched for alternative instruments that were customized to meet there needs, these options are called exotics. A lot of exotics are path-dependent that is; the payoff is dependent of the price evolution of the underlying assets and not only of the final price. Common path-dependent exotics are Asian, Barrier and look-back option. (Zhang, pp. 4-7)

An Asian option is based on the average of the underlying asset price. The average can be arithmetic or geometric where the arithmetic mean is calculated as:

$$\frac{1}{n} \sum_{i=1}^{n} a_i, \qquad i = 1, 2, \ldots, n$$

and the geometric mean:

$$\left(\prod_{i=1}^{n} a_i \right)^{1/n}, \qquad i = 1, 2, \ldots, n$$

The Asian option can be an average strike; the mean of the underlying replaces the strike price, or an average price; the mean replaces the final price. (Zhang, pp. 111-112)

A Barrier option is either a knock-in or a knock-out option, where the value of the underlying asset reaches a certain level the option comes into existence (knock-in) or ceases to exist (knock-out). A Barrier can be either a down-and-in, down-and-out, up-and-in or up-and-out option. The

simplest type is either European call or put, but there are several more complex types like time dependent Barrier options, Asian Barrier options etcetera. (Zhang, pp. 201-202)

A look-back option is an option depending on the maximum or minimum value of the underlying asset, the minimum value for a call option and the maximum value for a put option. The most common used look-back options are floating-, fixed strike look-back and partial look-back. For a floating look-back the strike price is exchanged for the maximum or minimum value, and for a fixed look-back the final price is exchanged for the maximum or minimum value. A disadvantage with floating and fixed look-back is that they are very expensive as they have a high premium. An alternative is therefore a partial look-back that only uses a percentage of the extreme value. (Zhang, p. 333)

Another type of exotic is a basket option whose payoff depends on the value of several underlying assets, where the assets normally are correlated stocks, stock indices or currencies. (Hull J. C., p. 541)

3.1.2   Index Linked Notes

An index linked note is a bond that is a debt instrument. The issuer of a bond owns the holder a debt (the principal amount) that has to be repaid at a maturity date. Besides the principal amount the issuer often must pay the holder an interest rate; a coupon that depends on the credit risk, the higher the credit risk the higher the coupon. A bond without interest rate is called a zero-coupon bond and is considered to be risk-less.

An index linked note is a bond type whose return depends on the performance of a stock index. The index linked note can be seen as a zero-coupon bond compounded with a call option.

3.2   Valuing derivatives

When valuing derivatives there are several aspects one must consider and understand.

3.2.1   Stochastic processes

A stochastic process is a process whose value is unpredictable over time. A Wiener process, or with another word Brownian motion is a process where the mean (drift rate) is zero and the variance rate is 1.0. A variable z following a Brownian Motion has two properties; the change of the variable during a small interval of time is:

$$\Delta z = \varepsilon \sqrt{\Delta t} \ \ \varepsilon \sim N(0,1)$$

and the values of z at any different time are independent.

A generalized Wiener process has unlike a Wiener process a drift rate. The process for a variable x is:

$$dx = a \, dt + b \, dZ$$

where a and b are constants and dZ is a Wiener process. The stock price is also assumed to follow a stochastic process, where the drift and variance rate of the stock price depend on the value of the stock:

$$dS = \mu S dt + \sigma S dZ$$
$$\Rightarrow \frac{dS}{S} = \mu dt + \sigma dZ$$

where S is the stock price, μ the expected rate of return per year and σ is the volatility of the stock price per year. This process is called a Geometric Brownian Motion (GBM). (Hull J. C., pp. 263-270)

3.2.2 Itô:s Lemma

Itô's Lemma is a very important theorem. If the stock price S follows a GBM, Itô:s Lemma shows that if G is a function of S and t, G is following the process:

$$dG = \left(\frac{\partial G}{\partial S}\mu S + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}\sigma^2 S^2\right)dt + \frac{\partial G}{\partial S}\sigma S dZ$$

If G is set to the logarithm of S:

$$dG = d\ln S = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma dZ$$

Since both the drift and variance rate are constants it means that lnS follows a generalized Wiener process and lnS is therefore normally distributed, which implies that the stock price has a lognormal distribution:

$$\ln S_T \sim N\left(\ln S_0 + (\mu - \frac{\sigma^2}{2})T, \sigma\sqrt{T}\right)$$

3.2.3 Black-Scholes pricing model

Fischer Black, Myron Scholes and Robert Merton have developed a very famous model for pricing options. According to Black and Scholes (pp. 640-641) it is necessary to assume some ideal condition to derive their formula:

- The stock price is following a GBM
- The short-term interest rate is known and constant
- There are no dividends
- There are no transaction costs
- Short selling of securities are permitted
- The derivative is European
- The valuation is done in a risk-free world (Appendix II)

Under these conditions Hull (p. 291) shows that it is possible to create a portfolio of a stock (S) following a GBM and a European derivative (f) that is a function of S and the time. The result is called the Black-Scholes-Merton's (B-S-M) differential equation ant it must be satisfied for all derivatives where the underlying value follows a GBM, otherwise there exist arbitrage opportunities:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2 = rf$$

From the B-S-M partial equation and together with boundary condition for a European call option, Black and Scholes (p. 644) have developed the Black-Scholes formula. With some modifications Black and Scholes formula can be used to price other derivatives, like stock indices, Asian options with geometric average and Barrier options where the barrier is checked continuously. For formulas see Appendix II.

4 Theoretical frame of reference

This chapter presents the theories that are; the principles behind standard MC simulation with pseudo numbers and QMC simulations. The chosen variance reduction techniques will also be presented.

4.1 Monte Carlo simulations

According to Glasserman (p. 1) the basics of standard MC methods are the relationship between probability and volume. MC methods use random numbers to estimate the volume of an integral. An integral over a unit interval can be written:

$$I = \int_0^1 f(x)dx$$

By interpreting the integral as the expected value $E[f(U)]$ we get a stochastic process, where U is a uniformly distributed number between 0 and 1. The expected value can be calculated by using independent random numbers U_i; $i = 1,..,n$ and the integral is approximated to a sum:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^n f(U_i)$$

The strong law of large numbers says that $\lim_{n\to\infty} \hat{I}_n = I$ with the probability one.

4.1.1 Standard error

One of the key features of MC simulation is the simple calculation of the standard error. By using confidence interval it is possible to get the accuracy of the simulation. The standard error can be derived from the expected value of the derivative:

$$E[\hat{X}_n] = X$$

where

$$\hat{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$$

with $E[X_i] = X$ and $Var[X_i] = \sigma_X^2 < \infty$. The central limit theorem says that when $n \to \infty$, the distribution

$$\frac{\hat{X}_n - X}{\sigma_X/\sqrt{n}} \sim N(0,1)$$

holds, which is equivalent with $\sqrt{n}(\hat{X}_n - X) \sim N(0, \sigma_X^2)$.

The real standard deviation σ_X is not observable but can be replaced with the standard deviation of the sample s_X, because $s_X/\sigma_X \to 1$ when $n \to \infty$. The standard error of the simulation becomes:

$$s_X/\sqrt{n}$$

This result shows that the convergence rate of the MC simulation is $O(1/\sqrt{n})$. To halve the error, four times as many samples are needed. Glasserman (p. 3) says that MC methods seldom are the best way to value one-dimensional integrals since the simulation is quite slow, but regarding multidimensional integrals it can be very useful as the convergence rate $O(1/\sqrt{n})$ still holds.

4.2 Pseudorandom number

A stochastic model is dependent of random numbers. In the computer there is no program for generating completely random number, but it is possible to use algorithms to create pseudorandom numbers that resembles of real random numbers. The generated numbers often come from a simple recursion, $x_n = g(x_{n-1})$. Even though the numbers are deterministic and not independent it has been shown that the difference between a pseudorandom and a completely random sequence is very small. (McLeish, p. 78)

Glasserman (p. 42) points out some important features when constructing a random number generator. *The period length;* all random sequences based on recursion will eventually repeat itself, with a longer period length this occurs more seldom. *Reproduction;* the possibility to reproduce a sequence is important for debugging. *Speed,* the algorithm must be fast since it can be called several times during one simulation. *Randomness,* the sequence must emulate a real random sequence. *Portability,* the same sequence should be generated on all computer platforms.

One of the most common types of random generators is the congruential generator, which has the form:

$$x_{n+1} = (ax_n + c) \bmod m$$
$$\varepsilon_n = x_{n+1}/m$$

where the multiplier a, modulus m and increment c all are positive integers chosen in advanced. The choice of the parameters is important for the randomness of the sequence. The recursion is initiated with the seed x_0. If c is zero the generator is called a linear congruential generator and otherwise a mixed congruential generator. The maximal period of the generator is $m - 1$ and for both generators there exist recommendation how to choose the parameters to get a maximal period.

Sometimes it is useful to split an unrelated random number sequences in several and this can be done by initiate the random number generator with different seeds (Glasserman, 2004). Important is that the seeds are distinct and for a linear congruential generator it is easy to assure that the random sequences differ. If

$$x_{n+1} = ax_{n+1} \bmod m$$

which implies

$$x_{n+k} = a^k x_{n+k} \bmod m$$

which is equivalent with:

$$x_{n+k} = (a^k \bmod m) x_k \bmod m$$

Then it is enough that $a^k \bmod m$ is calculated once and afterwards it is possible to construct a sequence that is *k* steps away.

4.2.1 Correlated random number

Creating correlated random number can be done with Cholesky factorization of the covariance matrix. (Haugh, pp. 5-6). According to linear algebra a symmetric positive definite matrix may be written as; $M = U^T D U$, where U is the upper triangle and D the diagonal.

From a sequence of random numbers $Z = Z_1, \ldots, Z_n$ (where $Z \sim N(0, I)$ and I is the unit matrices), a new sequence of random number $X = X_1, \ldots, X_n$ can be generated; $X = \mu + AZ$. The linear transformation property says that any linear transformation of a normal vector is again normal, that is:

$$X \sim N(\mu, \Sigma) \Rightarrow AX \sim N(\mu, A\Sigma A^T)$$

If $X = \mu + AZ$ then $X \sim N(\mu, AA^T)$ and $AA^T = \Sigma$. To generate X from Z we are searching for the parameter A which is the lower triangle of the covariance matrix. The Cholesky factorization of the covariance matrix result in

$$\Sigma = U^T D U = (U^T \sqrt{D})(\sqrt{D}U) = (\sqrt{D}U)^T (\sqrt{D}U) \Rightarrow A = \sqrt{D}U$$

By multiplying the Z with A, we get a new series with random numbers.

4.2.2 Generating normal distribution

In finance the most common distribution is the normal distribution, therefore it is needed to generate random numbers with this distribution. The simplest algorithm is called Box-Muller and is based on the properties of standard normal distributed variables N(0,1). If Z_1 and Z_2 are standard normal distributed variables then it exist a connection between them:

$$R = Z_1^2 + Z_2^2$$

This implies that the (Z_1, Z_2) is uniformly distributed on a circle with radius $\sqrt{R}$. If Z is unknown, it is possible to find Z by starting to generate R, with the formula $R = -2lnU_1$, where U_1 is uniformly distributed at the unit interval. Generating a random point on a circle (Z_1, Z_2) may be done by first generating an angle $V = 2\pi U_2$ (U_2 is also uniformly distributed at the unit interval) and map this angle to the point (Z_1, Z_2). Then the point will have the coordinates; $Z_1 = \sqrt{R}\cos V, Z_2 = \sqrt{R}\sin V$.

4.3 Variance Reduction Techniques

One of the disadvantages with MC simulation is that to get an accurate result a large number of trials are needed, which is expensive in terms of computer time. To increase the efficiency of the MC simulation there are different techniques that reduces the variance of the simulation.

4.3.1 Control Variate Technique

The control variate technique is one of the most effective and used techniques to increase the efficiency of MC. Hull and White (s. 243) shows that the technique is applicable when an option Y is valued with a numerical procedure, and there exist another derivative X similar to Y that has an analytical solution. The key feature of the technique is that the same MC simulation is used to calculate both the value of the derivative Y and the derivative X.

$$\hat{V}_Y^* = V_X + (\hat{V}_Y - \hat{V}_X)$$

with $\hat{V}_Y$ and $\hat{V}_X$ as the estimated values of Y and X, and V_X is the analytical solution of X.

If Y and X are unbiased and highly correlated it implies that the standard error of the control variate technique is less than the standard error of the simulation of Y. The standard error can be written as:

$$[\sigma_Y^2 + \sigma_X^2 - 2\rho_{XY}\sigma_Y\sigma_X]^{1/2}$$

which is less than σ_Y if:

$$\rho_{XY} > \frac{\sigma_X}{2\sigma_Y}$$

4.3.2 Antithetic Variate Technique

This method calculates two values of the derivatives. The first simulation is done with standard MC with normal distributed random number, ε, and the second simulation is done by changing the sign of the random numbers, $- \varepsilon$. The value of the derivative is calculated by taking:

$$\frac{V_1 + V_2}{2}$$

The standard error is measured the same way, with the average standard deviation of the simulation:

$$\frac{S_{AVG}}{\sqrt{n}} < \frac{s}{\sqrt{2n}}$$

Normally this standard error is less than the standard error calculated when using 2n simulations.

4.4   Quasi Monte Carlo

Quasi Monte Carlo (QMC) is based on the same technique as MC but the difference is that instead of using pseudo random numbers QMC tries to uniformly fill up the unit hypercube. The integral:

$$E[f(U_1, \dots, U_d)] = \int_{[0,1)^d} f(x)dx$$

is approximated with the sum

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

where $x_1, \dots, x_n$ are deterministically and carefully chosen points in the unit hypercube.

A difference between MC and QMC is that QMC methods have a large dependency of the dimensions. In standard MC if having a multidimensional problem it does not affect the random numbers. In QMC the dimension must be identified before points can be generated.

4.4.1   Net, sequences and discrepancy

The discrepancy measures the deviation from uniformity, (Glasserman, pp. 283-284) the lower discrepancy the better the uniformity. If $x_1, \dots, x_n = I^d$ and $\mathcal{A} \in I^d$, where $\mathcal{A}$ is a collection of all rectangles in the unit hypercube:

$$\prod_{j=1}^{d}[u_j, v_j), \quad 0 \le u_j < v_j \le 1$$

the discrepancy of $x_1, \dots, x_n$ can be defined as;

$$D(x_1, \dots, x_n; \mathcal{A}) = \sup_{A \in \mathcal{A}} \left| \frac{\#\{x_i \in A\}}{n} - \lambda(A) \right|$$

where $\#\{x_i \in A\}$ denotes the number x_i in A and $\lambda(A)$ is the Lebesgue measure of A, (the volume).

By restricting $\mathcal{A}$ to rectangles on the form:

$$\prod_{j=1}^{d}[0, u_j)$$

we get the star discrepancy D^*. If the sequence is a low discrepancy sequence the star discrepancy converges with the rate:

$$D^*(x_1, \dots, x_n) = O\left(\frac{(\log n)^d}{n} \right)$$

which can be approximated with $O(1/n^{1-\epsilon})$ where $\epsilon > 0$ and this convergence rate is mostly faster than $O(1/\sqrt{n})$.

One of the best ways to create a low discrepancy sequence is by using *(t, m, s)-nets* and *(t ,s)-sequences*. Niederretier at al present (pp. 58-60) the theory. An elementary interval E (a subset) in $[0,1)^s$ in base $b \geq 2$ may be written as:

$$E = \prod_{i=1}^{s} \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right)$$

with integers, $d_i \geq 0$ and $0 \leq a_i \leq b^{d_i}$ for $1 \leq i \leq s$.

Letting $0 \leq t \leq m$ be integers, a *(t, m, s)-net* in base b is a sequence of b^m points in $[0,1)^s$. Every elementary interval E of volume b^{t-m} contains exactly b^t points. A sequence of $x_1, x_2, ...$ in $[0,1)^s$ is a *(t, s)-sequence* in base b if for all $m > t$ each segment $\{x_i : jb^m < i \leq (j+1)b^m\}$ where $j = 0,1, ...,$ is a *(t, m, s)- net* in base b.

Figure 2 illustrate an example (Li & Mullen, p. 4) of a (0,2,2) net in base 3, that is t=0, m=2, and s=2, meaning that the net contains $b^m = 3^2 = 9$ points, and every elementary interval of volume $b^{t-m} = 3^{0-2} = 1/9$ contains $b^t = 3^0 = 1$ point.
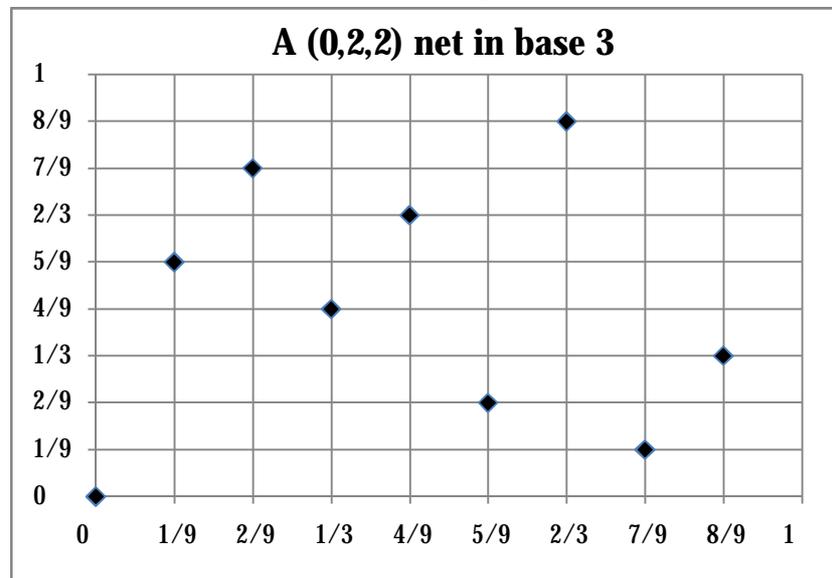
Figure 2. A (0,2,2) net in base 3

The discrepancy measure is important when calculating the standard error for a sequence (Boyle & Tan, p. 11) this result is called Koksma-Hlawka inequality and bounds the integration error by using the star discrepancy and the integrand:

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \int_{[0,1]^d} f(u)du \right| \leq V(f)D^*(x_1, ..., x_n)$$

where f is a function with bounded variation $V(f)$. The Koskma-Hlawka separates the integration error in two different terms, the roughness of the integrand and the uniformity of the sequence. But even though the result of Koskma-Hlawka is interesting in theory, it is difficult to use; both the variation and the star discrepancy are difficult to compute. If the parameters are known there are still disadvantages with Koskma-Hlawka though the integration error often is overestimated.

Glasserman (p. 325) suggest a way to measure the error of a low discrepancy sequence by using the root mean square error and the root mean square relative error.

$$RMSE(n) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{C}_{i(n)} - C_i\right)^2}$$

$$RMSRE(n) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{C}_{i(n)} - C_i}{C_i}\right)^2}$$

where $\hat{C}_{i(n)}$ is the approximated value, C_i is the true value and m is the number of problems.

4.4.2 Van der Corput

Van der Corput is a one-dimensional low discrepancy sequence, where every integer k can be represented as a linear combination in the base $b \geq 2$.

$$k = \sum_{j=0}^{\infty} a_j(k)b^j \quad (4.1)$$

The radical inverse function ψ_b maps every number k, written in base b, to a decimal point in the interval [0, 1):

$$\psi_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}} \quad (4.2)$$

The four first points in base two of a Van der Corput sequence are produced like:

$$k = 1 = 1 * 2^0 \Rightarrow \psi_2(1) = \frac{1}{2^1} = \frac{1}{2}$$
$$k = 2 = 1 * 2^1 + 0 * 2^0 \Rightarrow \psi_2(1) = \frac{1}{2^2} + \frac{0}{2^1} = \frac{1}{4}$$
$$k = 3 = 1 * 2^1 + 1 * 2^0 \Rightarrow \psi_2(1) = \frac{1}{2^2} + \frac{1}{2^1} = \frac{3}{4}$$
$$k = 4 = 1 * 2^2 + 0 * 2^1 + 0 * 2^0 \Rightarrow \psi_2(1) = \frac{1}{2^3} + \frac{0}{2^2} + \frac{0}{2^1} = \frac{1}{8}$$

The Van der Corput sequence is a sequence in one dimension, but it is the key element for multidimensional low discrepancy sequences.

4.4.3 Halton

The foundation of a Halton sequence is the Van der Corput sequence. Halton uses distinct prime bases $b_1, \ldots, b_d$, where b_1 is chosen to be the lowest prime (two) and so on. The radical inverse function is defined as in (4.2) and the decimal point of the number k may be written:

$$x_k = \left(\psi_{b_1}(k), \ldots, \psi_{b_d}(k)\right), k = 0,1,2, \ldots$$

Table 1 represent the eight first numbers in base two and three in a two dimensional Halton sequence where $\psi_2(k)$ is the Van der Corput sequence.

K	$\psi_2(k)$	$\psi_3(k)$
0	0	0
1	1/2	1/3
2	1/4	2/3
3	3/4	1/9
4	1/8	4/9
5	5/8	7/9
6	3/8	2/9
7	1/16	8/9

Table 1. The eight first numbers in base two and three in a Halton sequence

The 1000 first point plotted in a Halton sequence in dimensional two can be seen in Figure 3.
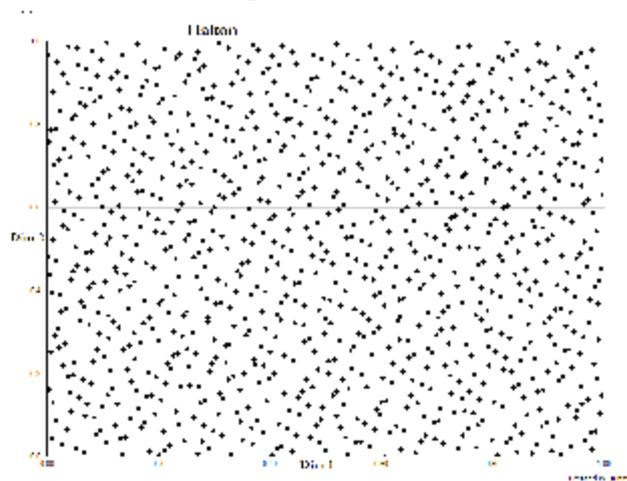

Figure 3. The 1000 first point in base 2 and 3 in a Halton sequence, plotted in Quantlab

The Halton sequence suffers from correlation problems in higher dimensions, and the points do not have a uniformly distribution. This can be seen in the Figure 4.
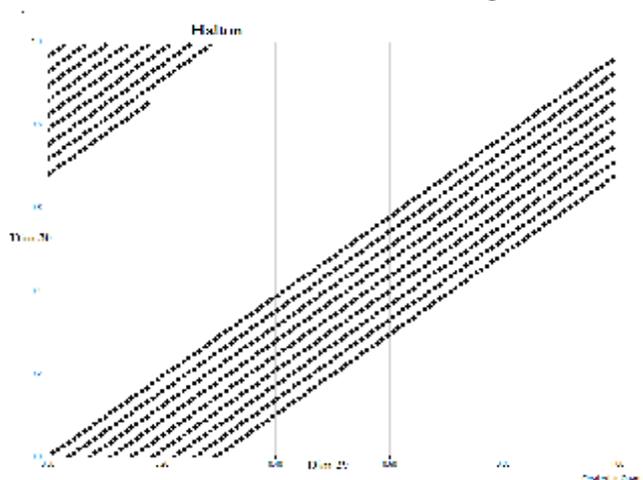

Figure 4. The 1000 first points in base 29 and 31, in a Halton sequence, plotted in Quantlab

4.4.4 Faure

Faure is a *(0,s)-sequence* in base b that starts from the Van der Corput (equation 4.1). The Faure sequence differs from the Halton sequence by using a common base for all dimensions. The base is at least as large as the dimension and it must be a prime. Every number k, in dimension $i = 1, 2, \dots, d$, is permuted to the interval [0,1):

$$\sum_{j=0}^{\infty} \frac{y_j^{(i)}(k)}{b^j}$$

where

$$y_j^{(i)}(k) = \sum_{l=0}^{\infty} \binom{l}{j-1} (i-1)^{l-j+1} a_l(k) \bmod b \quad (4.3)$$

with

$$\binom{m}{n} = \begin{cases} m!/(m-n)!\,n!\,, & m \geq n \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

The base b representation of k has exactly r number of bits, the sum in equation 4.3 is finite and all numbers $l \geq r$ is zero. The condition 4.4 says that if $m \leq n$, then the equation 4.4 is zero which also implies that the equation 4.3 is zero when $r + 1 \leq j$. The equation 4.3 has maximum r terms that is not zero and can be written:

$$\begin{pmatrix} y_1^{(i)}(k) \\ y_2^{(i)}(k) \\ \vdots \\ y_r^{(i)}(k) \end{pmatrix} = \mathbb{C}^{(i-1)} \begin{pmatrix} a_0(k) \\ a_1(k) \\ \vdots \\ a_{r-1}(k) \end{pmatrix} \bmod b$$

where

$$\mathbb{C}^{(i)}(m,n) = \binom{n-1}{m-1} i^{n-m}$$

for $n \geq m$, otherwise 0. The matrix $\mathbb{C}^{(i)}$ has cyclic properties so $\mathbb{C}^{(i)} = \mathbb{C}^{(1)}\mathbb{C}^{(i-1)}, i = 1,2,\dots,$. This cyclic property has implications for higher dimensions, where several dimensions produce the same sequence. This can be seen in Figure 5.
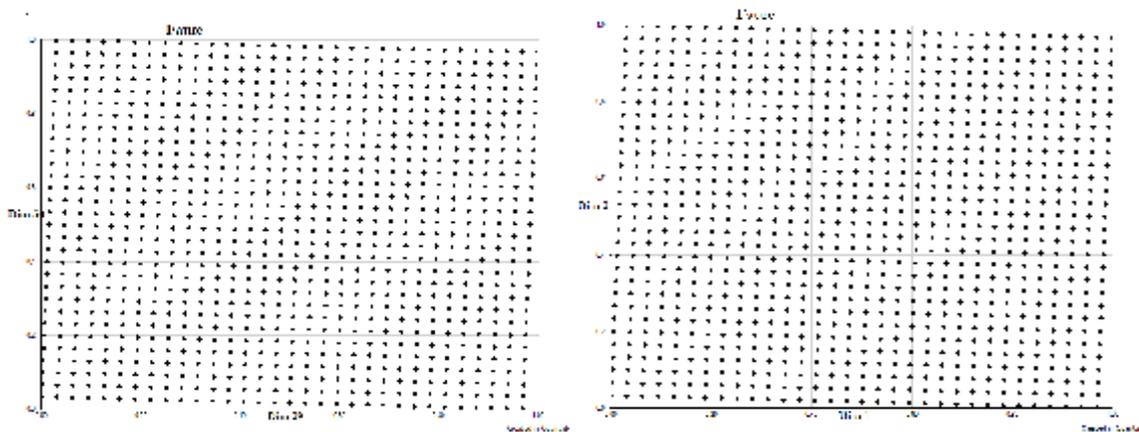

Figure 5. The first 1000 points in the Faure sequence in base 31, the left picture plot dimension 29 and 30 and the right picture plot dimension 1 and 2 and both of the pictures are identical.

Table 2 illustrates the twelve first numbers in the Faure sequence of dimension three, which means that base three is used for every dimension. Here it can be seen that every dimension

produces a similar sequence, a suggestion to improve the uniformity is to start the sequence on point $b^4 - 1$.

Number/Dimension	1	2	3
1	1/3	1/3	1/3
2	2/3	2/3	2/3
3	1/9	4/9	7/9
4	4/9	7/9	1/9
5	7/9	1/9	4/9
6	2/9	8/9	5/9
7	5/9	2/9	8/9
8	8/9	5/9	2/9
9	1/27	16/27	13/27
10	10/27	25/27	22/27
11	19/27	7/27	4/27
12	4/27	19/27	7/27

Table 2. The thirteen first numbers in dimension (1-3) in the Faure sequence, with base 3

4.4.5 Sobol

The basic of the Sobol sequence is also the Van der Corput sequence. A Sobol sequence is a *(t,s)-sequence* in base two which means that every number k is possible to write as a binary representation:

$$k = a_0(k) + 2a_1(k) + \cdots + 2^{r-1}a_{r-1}(k)$$

where a_i is zero or one and r is the number of bits in the k: th binary representations.

Bratley and Fox (pp. 89-93) present how to implement the Sobol sequence. First a set of direction number is needed that can be written as:

$$v_i = 0.v_{i1}v_{i2}v_{i3} \ldots \quad i = 0, \ldots, r$$

where v_{ij} is either 0 or 1 and represent the j:th bit in the expansion of v_i and the direction number v_i can be expressed like:

$$v_i = \frac{m_i}{2^i}$$

To find the number m_i, it is necessary to choose a primitive polynomial of degree q:

$$x^q + c_1 x^{q-1} + \cdots + c_{q-1}x + 1$$

where each coefficient c is represented by a 0 or 1. Table 3 presents the ten primitive polynomials.

Degree	Primitive polynomial		
0	1		
1	$x + 1$		
2	$x^2 + x + 1$		
3	$x^3 + x + 1$	$x^3 + x^2 + 1$	
4	$x^4 + x + 1$	$x^4 + x^3 + 1$	
5	$x^5 + x^2 + 1$	$x^5 + x^3 + 1$	$x^5 + x^3 + x^2 + x + 1$

Table 3. Ten primitive polynomials, there exist three further polynomials of degree five

The first dimension uses the first polynomial etcetera, for higher dimension when there are several polynomials to choose between the polynomial can be chosen randomly. When the primitive polynomial is chosen it is possible to define a recurrence to calculate m_i:

$$m_i = 2c_1 m_{j-1} \oplus 2^2 c_2 m_{j-2} \oplus \cdots \oplus 2^{q-1} c_{q-1} m_{j-q+1} \oplus 2^q m_{j-q} \oplus m_{j-q} \ (4.5)$$

where $m_1, \ldots, m_q$ must be initiated. The initializing can be done freely as long as m_i is an odd integer and inside the interval $0 < m_i < 2^i$. $\oplus$ is the bit-by-bit exklusive-or (XOR) operator. Finally, the representation x_k may be expressed as a binary vector:

$$x_{k=} a_0(k)v_1 \oplus a_1(k)v_2 \oplus \cdots \oplus a_{r-1}(k)v_r$$

and thereafter calculating the binary representation of x_k.

Antonov and Saleev (pp. 252-256) have made some modification in Sobol's original method, and propose that instead of using binary representation of k, the Gray code can be used. In a Gray code the number k and k+1 differ in only one bit, see Table 4.

	1	2	3	4	5	6	7	8	9	10
Binary	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
Gray code	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111

Table 4. The first ten numbers in binary- and Gray code representation

The Gray code is calculated from the binary representation of $k \oplus \lfloor k/2 \rfloor$. The bit that will be changed in the Gray code for k+1 is the rightmost zero bit in the binary representation of k. For example number two is represented as [1,0] in binary code and [1,1] in Gray code. The rightmost zero bit in the binary representation of two is the least significant bit, and therefore the Gray code of three will be [1,0]. That means that the Gray code representation of two [1,1] is only changed in one bit, see Table 4.

Finally the decimal representation of k can be expressed as:

$$x_{k=} g_0(k)v_1 \oplus g_1(k)v_2 \oplus \cdots \oplus g_{r-1}(k)v_r \ (4.6)$$

and if k and k+1 differ in the l:th bit (the right most zero bit in the binary representation of k) then

$$x_{k+1} = g_0(k+1)v_1 \oplus g_1(k+1)v_2 \oplus \cdots \oplus g_{r-1}(k+1)v_r$$
$$x_{k+1} = g_0(k)v_1 \oplus g_1(k)v_2 \oplus \cdots \oplus (g_l(k) \oplus 1)v_l \oplus \cdots \oplus g_{r-1}(k+1)v_r$$
$$x_{k+1} = x_k \oplus v_l \ (4.6)$$

which implies that x_{k+1} can be computed recursively from x_k.

There are some important remarks on the Sobol generator; the $m_1, \ldots, m_q$ must be initiated and even if they can be chosen freely there are some recommendation how to get the best uniform properties. Sobol provides some guidance concerning the properties of the initial number called Property A. Bratley and Fox suggest initial numbers together with polynomials up to 40 dimensions and Joe and Kue (Sobol Sequence Generator) have developed this further and suggest initial numbers up to 1111 dimensions, where the initial numbers are fulfilling property A.

To illustrate the Sobol sequence, let us say that we have three dimensions *d*=3 and we want to represent the numbers one to eight in the Sobol sequence, that means that the maximal number of bit; *r,* is four, therefore four direction number are needed. That means that we need four initial numbers m_i. For the first sequence the primitive polynomial is; 1, meaning that also the initial numbers $m_i, i = 1, \ldots, 4$ will be one which implies that the direction numbers will be:

$$v_1 = 0.1000, v_2 = 0.0100, v_3 = 0.0010, v_4 = 0.0001$$

The decimal representation of one will be calculated according the formula (4.6):

$$x_1 = 1 * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \oplus 0 * \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \oplus 0 * \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \oplus = 0 * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \frac{1}{2} + \frac{0}{4} + \frac{0}{8} + \frac{0}{16} = 0.5$$

and thereafter the recursion can be used meaning that the decimal representation of two will be represented as:

$$x_2 = x_1 \oplus v_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow x_2 = 0.75$$

since the right most zero bit of x_1 is the bit number two.

If instead calculate the values for the third sequence, that means that the primitive polynomial will be $x^2 + x + 1$, that is a polynomial of degree two, which implies that we need two initial number, chosen to be; $m_1 = 1$ and $m_2 = 1$ and m_3 is generated according to the formula (4.5):

$$m_3 = 2c_1 m_2 \oplus 2^2 m_1 \oplus m_1 = 2 \oplus 4 \oplus 1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = 7$$

m_4 is generated the same way and that gives $m_i = (1,1,7,11)$. Dividing m_i on 2^i, means that the binary point is shifted to the left i places, and that gives $v_3 = 0.1110$. To calculate the decimal point x_3 the right most zero bit in x_2 is identified to the first bit, and that implies that the decimal point is calculated:

$$x_3 = x_2 \oplus v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow x_3 = 0.25$$

4.4.6 Generating normal distribution

Box Muller is not a good method to use for a low discrepancy sequence since it will affect the properties of the sequence. Instead a method called Moro's inverse can be used. Beasley and Springer (Glasserman, p. 67) have created an inverse function:

$$\phi^{-1}(u) \approx \frac{\sum_{n=0}^{3} a_n \left(u - \frac{1}{2}\right)^{2n+1}}{1 + \sum_{n=0}^{3} b_n \left(u - \frac{1}{2}\right)^{2n}}$$

The inverse function is symmetric $\phi^{-1}(1 - u) = -\phi^{-1}(u)$ hence it is enough to approximate the upper interval [0.5, 1). Beasly and Springers approximation is used for $u < 0{,}92$ and then Moro has done some modification of the tail:

$$\phi^{-1}(u) \approx \sum_{n=0}^{8} c_n \lfloor \ln(-\ln(1 - u)) \rfloor^n$$

Where a_n, b_n and c_n are given values.

4.5   Generating path

When generating paths for an European derivative, dependent on one variable S, the procedure is according to Hull (p. 411) as following:

- Sample a random path for S in a risk-neutral world
- Calculate the expected payoff at maturity date
- Repeat the first two steps, until you have enough with samples
- Calculate the mean of the payoffs to get the expected payoff
- Discount the mean with the risk-free rate to get the value of the derivative

If the market variable S follows a GBM, it has a lognormal distribution and the path will be sampled from the formula:

$$S(T) = S(0)\exp\left[\left(r - q - \frac{\sigma^2}{2}\right)T + \sigma\varepsilon\sqrt{T}\right](4.7)$$

In the case when the payoff is path-dependent then it is necessary to split the equation 4.7 into several time steps.

$$S(t_{i+1}) = S(t_i)\exp\left[\left(r - q - \frac{\sigma^2}{2}\right)(t_{i+1} - t_i) + \sigma\varepsilon_{i+1}\sqrt{(t_{i+1} - t_i)}\right]$$

5 General framework

The main purpose with the Master's thesis is to develop a general framework. This chapter analyse the different methods and the best way to generate a general framework. Thereafter it presents the chosen way to implement the framework and test it with the Smart Start.

5.1 Random numbers

The random numbers that will exist in the general framework are Sobol and pseudo. Table 5 compares the uniformity properties of the sequences in different dimensions; the coloured areas are the areas performing best for each dimension. It can be seen in the table, that it is mostly the Sobol sequence that performs best.

Comparison U[0,1] N=1000												
	Theoretic	**Pseudo**	**FAURE**			**SOBOL**			**HALTON**			
			Dim 2	Dim 15	Dim 30	Dim 2	Dim 15	Dim 30	Dim 2	Dim 15	Dim 30	
			Base 3	Base 17	Base 31	Base 2	Base 2	Base 2	Base 3	Base 47	Base 113	
Mean	0,50000	0,50106	0,50029	0,49786	0,49788	0,49978	0,49958	0,50034	0,49852	0,48954	0,48951	
kurtosis	-1,20000	-1,12569	-1,20131	-1,19990	-1,20412	-1,20161	-1,19917	-1,20140	-1,19938	-1,21182	-1,18264	
skew ness	0,00000	0,02740	0,00188	-0,00359	0,01583	0,00149	0,00130	0,00188	0,00171	0,02176	0,01631	
variance	0,08333	0,07991	0,08328	0,08334	0,08325	0,08332	0,08318	0,08328	0,08335	0,08382	0,08156	
Max	1,00000	0,99938	0,99902	0,99715	0,99896	0,99902	0,99902	0,99902	0,99863	0,98778	0,99170	
Min	0,00000	0,00025	0,00098	0,00020	0,00104	0,00098	0,00098	0,00098	0,00046	0,00045	0,00008	

Table 5. Comparison of the uniformity of the different sequences

The literature points out that the Halton sequence is not a *(t,s)-sequence* or a *(t,m,s)-net*, and it is also obvious that the sequence suffers from high correlation already in early dimensions and it is not filling up the unit interval. The Faure sequence will not be used. It is not built-in into the function library in Quantlab and it is not efficient as it is implemented now. The Faure sequence also seems to perform worse than Sobol, but this is not examined enough to draw a conclusion. The Faure produces the same sequence several times and a suggested way to improve the uniformity is to start the sequence with a later number. But when looking at the sequences in Figure 6 with the left picture starting on $b^4 - 1$ and the right picture starting on zero, it is not obvious that the left picture show more uniform properties than the right.
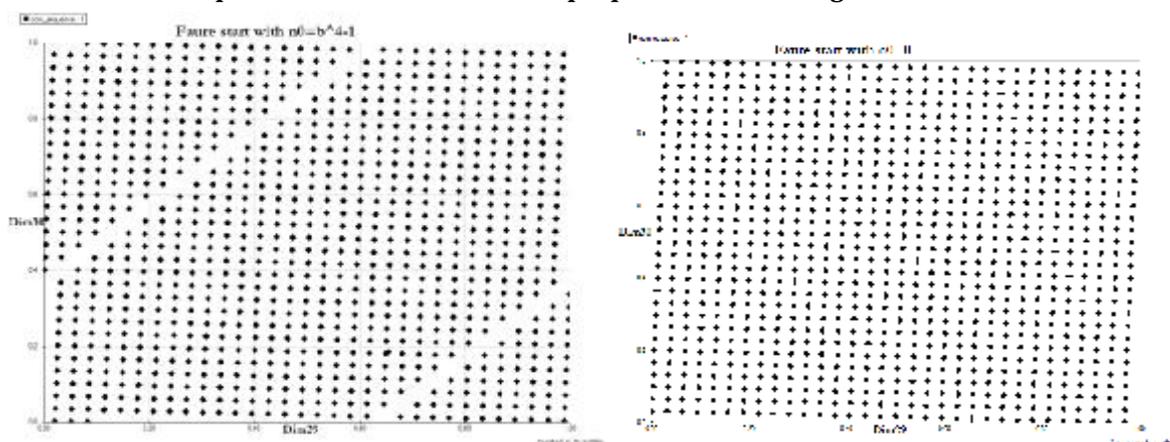

Figure 6. The 1000 first points in the Faure sequence, where the left picture start at point b⁴-1 and the right picture at point 0.

If valuing a path dependent instrument the number of time steps are the number of dimensions. For every sample a vector of the dimensions is needed, and in early samples this

vectors will consist of identical numbers. Example of this can be seen in Table 6. From this table it is obvious that it is better to start the sequence with a later number.

Sample/Dimension	1	2	3	4	5	6	7	8	9
1	0.091	0.091	0.091	0.091	0.091	0.091	0.091	0.091	0.091
2	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182
3	0.273	0.273	0.273	0.273	0.273	0.273	0.273	0.273	0.273
.									
998	0.750	0.700	0.014	0.874	0.097	0.866	0.089	0.675	0.808
999	0.841	0.791	0.105	0.965	0.187	0.956	0.180	0.766	0.89

Table 6. The Faure sequence constructed with base 11

The same principle is used for the Sobol sequences, the first trials produce pretty much the same numbers and it is better to start with a later number. Visually it seems like that when starting the Sobol sequence with a later number it also produces a more uniform sequence, see Figure 7
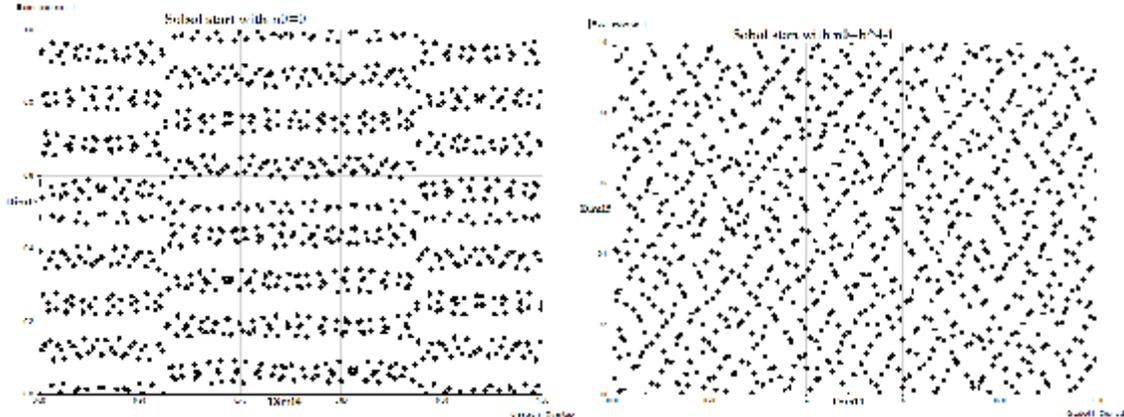

Figure 7. The 1000 first points in the Sobol sequence, the left picture starting on point 0 and the right with a later number.

The random number will for every sample return a vector of the size dimension. When valuing a portfolio or a basket where the assets are correlated, the random numbers are created by the Cholesky factorization. Here the requirement is that the user must specify the covariance matrix as an input, and the matrix must be positive definite. This is not always the case so a potential problem may arise if the matrix is only positive semidefinite. This could be solved by controlling the matrix, and in the case where the matrix is semidefinite reduce the rows and columns.

5.2 Variance reduction techniques

The user can choose standard MC or antithetic variable. The control variate technique is not implemented; the user must implement the tecnhique in the payoff function. The control variate technique may use the same vector with the underlying values to calculate both the instrument and the control variable or the user can implement the control variable and standard variable as two assets and use the same random numbers when simulating them. Since the control variable requires an analytic value it should be very difficult to implement in a general framework.

5.3 Function pointers

As a solution to generalize the framework, function pointers with object will be used, because they let the user specify the functions by themselves. It is too complicated to specify the specific functions in the framework, and separating them with if-statements, because there exist so many

possibilities, and the framework would never cover them all. If using if-statements it would also require more input variables, with increases the complexity of the functions.

5.3.1 Non constant volatility and interest rate

Often instruments are having volatilities and interest rates that are non constants. It is desirable that the MC framework can handle this requirement, and since the framework does not know how the non constant parameters are calculated function pointers are used.

As input values there will be the number of the asset so when simulating multiply assets the user knows which asset that is calculated and if the volatilities for the assets are calculated in different ways it is easy for the user to separate them by using if-statement. Other input will be the random numbers, the previous volatility/interest rate, the time step and the object with the needed information to calculate the non constant parameters.

A remark on this solution is that the random number will be the same as the random numbers that simulate the path. If for example the volatility is expected to follow a Wiener process it is possible to believe that this process is not identical with the process for the path, and the random numbers should be generated independent of the random numbers that is used for simulating the path.

This would require that the framework generated several sequences with random numbers, and it would not be a problem when pseudo random numbers are used but as the Sobol numbers are very dependent on the dimension, it is not only to call the Sobol function once more to get the next sequence. The solution to generate several paths would also require extra input values that specify how many sequences that should be generated. Because the non constant parameters are not investigated very deeply in the thesis, the framework will as it is now, send in the same random numbers as for the path.

5.3.2 Simulating Path

The Master's thesis is delimited to generate a path that is following a GBM. But desirable is that the framework can handle different kinds of paths, like short rates etcetera. Of this reason the path generator will also be a function pointer. Right now the path that will be generated must have log normal properties, since the random numbers that are generated are normal distributed.

As input values for the path generator there is the previous underlying value, the volatility, the interest rate, the time step, the random number, the number of the asset and an generic object where the user can store constants etcetera needed for generating the path, for example the dividend for a asset following a GBM.

The Master's thesis cannot guarantee that these input values are enough for generating all kinds of path, for this further analyses are required and different stochastic processes must be studied further.

The path will be simulated for every sample where the simulation starts to calculate the value for the first dimension, thereafter the second and so on. If multiply assets the first asset will be simulated first and thereafter the second one.

Here a potential conflict arises when working with Sobol numbers, the dimension for a basket option is the number of stocks in the basket, and for a Barrier option it is the number of time step. It is difficult to combine these two definitions of the dimension, and since there may be a basket that exists of path-dependent instrument, no respect is taken to the fact that the dimension also can be the number of assets. A basket consisting of one dimensional assets will be valued the same as a basket consisting of multiple assets. This will affect the simulation because now all the assets will be simulated by using the first dimension indeed modified with a correlation coefficient, but the result would still be different if asset number one was simulated with dimension one, asset number two was simulated with dimension two and so on.

5.3.3 Stop Criterion

It is common that exotic instrument has a stop criterion, like for the knock-out option, where the function ceases to exist if a barrier is reach. This can of course also be start criterion like for the knock-in option where the function comes into existence when the barrier is reach. The stop criterion can be very different, for example if having several assets it can be a function of the assets; hence it is difficult to generalize this function and a function pointer is used.

The function will have some necessary inputs where the first is the simulated path of the instrument. This path will be sent into the function as a matrix of size (dimension+1)*assets and if only having one asset it is still a matrix. As input it will also be a second path representing the antithetic variate technique. If this technique not is used, the matrix will be of size zero.

Next input will be a matrix, initialized with zeros. This matrix will also be the output of the function. The matrix is of size 2*assets, where the second row represent the antithetic variate technique, and if it is not used the user does not have to care about it. The suggestion is that the user checks the stop criterion for each asset and changes the zeros to one if the criterion is reached, if there is no stop criterion the user can just return the matrix as it is.

The last input will be an object where the user can specify the stop criterions. To help the user to calculate the stop criterion there exist two functions:

- *up(vector (number) s, number b)* – returns 1 if the price reaches up to the barrier, else 0
- *down(vector (number) s, number b)* – returns 1 if the price goes down to barrier, else 0

5.3.4 Payoff functions

Different derivatives have different payoffs, and it is very difficult to create a function that can handle several instruments, even if the instruments are similar. The proposed solution to spare computer time and complexity is that the users identify and implement the payoff function by themselves; therefore a function pointer will be used, where the user implement the function in his own workspace.

Exactly like for the stop criterion the function will have the simulated path/paths as input. Next input will be a matrix of size 2*assets, that is the stop criterion matrix. It is important that the user is consequent in the use of the stop criterion matrix; it will be initialized with zeros but the framework does not control how the user handles it in the stop criterion function. With this matrix the user can calculate the payoff and knows whether an instrument ceases to exist or comes into existence.

Next two input parameters are the complete life time of the instrument and the interest rate matrix. This so the user can discount the value of the instrument. The last input parameter is an object containing the rest of the variables needed to calculate the payoff.

To help to calculate the payoff there exist some standard functions:

- *value_call(number k, number s)* - returns the value of a call option
- *value_put(number k, number s)* - returns the value of a put option
- *geometric_average(vector (number) s)* – returns the geometric average of a vector

Already implemented in the function library in Quantlab:

- *v_max(vector (number) v)* – returns the max value in the vector
- *v_min(vector (number) v)* – returns the min value in the vector
- *v_average(vector (number) v)* – returns the arithmetic average of a vector

5.4 Implementation of the general framework

To create a random framework implies that the complexity of the code increases and several functions must be specified of the user. The purposed solution will still have some requirement on the path that will be simulated and the input parameters.

5.4.1 Input functions and parameters

The code is implemented so it is possible to value an instrument consisting of multiply assets, for example a basket option. It is up to the user to define the input values for the MC simulation that consist of two input functions:

- *calc_values* – calculated the value of the instrument and returns one value
- *calc_values_conv* – calculates the convergence of the values, returns a vector. As an extra input value this function needs the step, that is how often to calculate a value

Thereafter there exist two possible ways how to call the functions, either the function has a

- Start date (A date)
- Final date (A date)
- The dimension (A number)

as input, or if the user before valuing the instrument is sure over the exact dates to check the instrument, the input is only the:

- Observation dates (A vector consisting of all the observation dates)

No matter which input function the user will choose, there exist some input values that always are needed and these are represented in the list:

- The number of trials
- The seed to the random number generator
- A vector consisting of the initial value of every instrument that will be simulated, the size of the vector is the number of assets
- A vector consisting of the initial volatility of every instrument, the size of the vector is the number of assets
- A vector consisting of the initial interest rate of every instrument, the size of the vector is the number of assets.
- The covariance matrix, (A matrix of size assets*assets), it is set to size zero if there is only one asset
- A MC simulation method (A string either standard or antithetic)
- A random number method (A string either Sobol or pseudo)

Into the four functions several function pointers are needed as input, together with them always an object will be needed, the function pointers are:

- *The volatility function,* returns a number, the input values are:
  - The asset number
  - The initial volatility
  - The volatility at the previous time step
  - The change in time
  - A generic object
- *The interest rate function,* returns a number, the input values are:
  - The asset number
  - The initial interest rate
  - The interest rate at the previous time step
  - The change in time
  - A generic object

- *The path function,* this function returns a number that is the current price
  - The asset number
  - The previous price
  - The volatility
  - The interest rate
  - The random number
  - The time step
  - A generic object
- *The stop function,* returns a matrix. The input values are:
  - A matrix of size 2*assets initialized with zeros
  - A matrix consisting of the simulated path of size dimension*assets
  - A matrix consisting of the second simulated path, if antithetic variate technique is used the vector is of size dimension *asset if standard MC it is of size 0
  - A generic object
- *The payoff function,* returns a value and has input values:
  - A matrix consisting of the simulated path, size dimension *assets
  - A matrix consisting of the second simulated path, if antithetic variate technique is used the vector is of size dimension *asset if standard MC it is of size 0
  - A matrix of size 2*assets that represent the stop criterion
  - The time to maturity
  - The interest rate matrix,
  - A generic object

Note that this order of the input parameters is not the correct order, it is only the needed input parameters.

5.4.2   Set the time step

The first that is done in the input function is to set the time step. It exist two different kinds of functions depending on choice of input function.

- *calc_time_step(date start_date, date final_date, number dimension)*
- *calc_time_step(vector (date) observation_dates)*

Both the functions are returning a vector with the time steps.

5.4.3   The random numbers

Since it is only pseudo random numbers respectively Sobol numbers that are built-in Quantlab, the user can only choose between these methods:

- *set_sobol(number dim, number seed)* & *set_pseudo(number dim, number seed)*
- *next_sobol()* & *next_pseudo()*

These functions returns a vector at the same size as the dimension, the set function sets the first vector and initialize the sequence, when the next function get the next vector. The methods; pseudo and Sobol are separated by using if statements and the user have to specify the method as a string.

5.4.4   Variance reduction techniques

The user must choose if using standard MC or antithetic variable, these methods are separated by if statements where the method is specified as a string. The antithetic variable technique will not return two matrices; instead it will return a matrix of double size, where the first half is the values calculated with the random numbers $,\varepsilon,$ and the second part of the vector has used the random numbers, $-\varepsilon$. This matrix is then split up into two sub matrices before sending it to the user.

If the user wishes to use the control variate technique, the user must implement it in the payoff function.

5.4.5   The path generator for a GBM

In the specific case when the instrument is following a GBM the function that generate the path looks like:

```
number path_GBM(number nb_a, number s, number t, number vol, number r, number rand, number q){
        return s*exp(((r-q)-vol^2/2)*t + vol*rand*sqrt(t));
}
```

Where s is the previous price, t is the time step, vol is the volatility, r is the interest rate, q is the dividend and rand is the random number.

5.4.6   Code hierarchy

The hierarchy of the framework is showed in Figure 8. First the chosen input function is called and the time step, the Cholesky factors are calculated and the first random numbers are set. Thereafter there is a loop over the number of samples that for every sample starts to set the volatility and interest rate, calculates the path and the value. A value is returned and stored in a vector, subsequently the next random number is set and the procedure is calculated again. When the loop has gone through all the number of trials, the values for every sample is stored in a vector, and the average value/values are calculated.
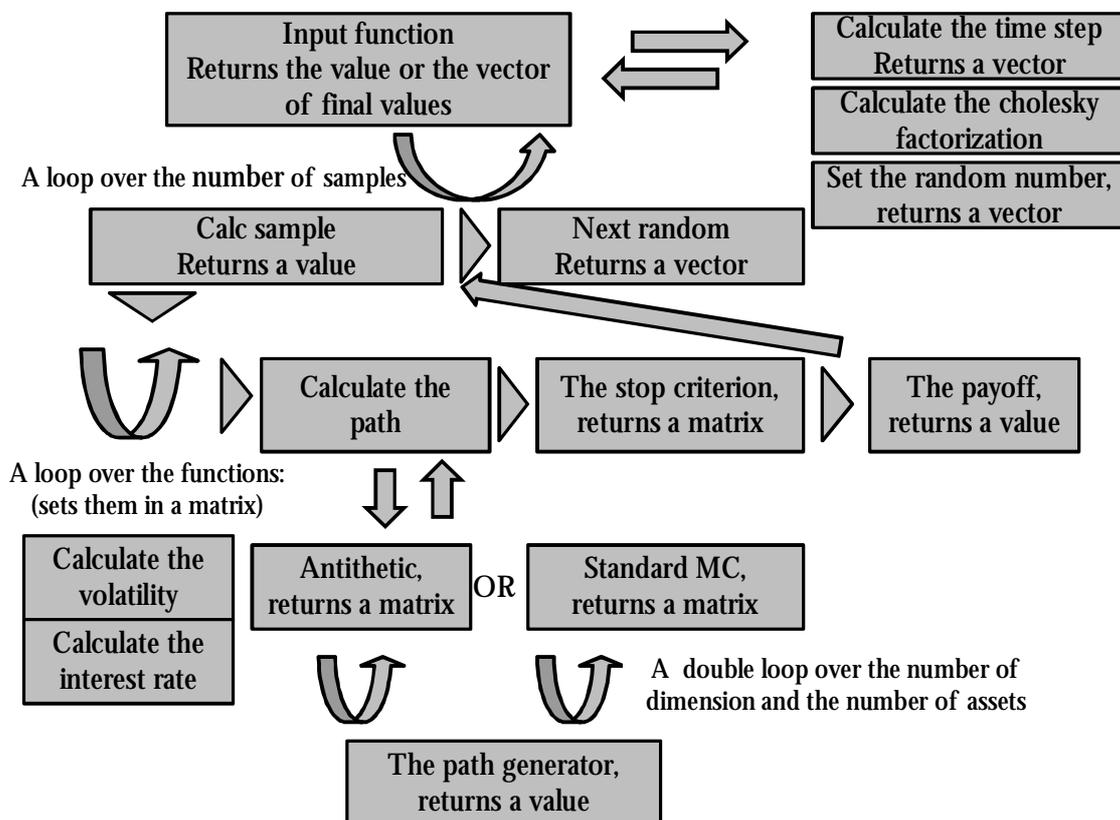

Figure 8. The code hierarchy

This code hierarchy allows the Smart Start to be simulated in the framework. The parameters as volatility, dividend, correlation etcetera were chosen to realistic numbers, and thereafter the framework was executed. No further analysis will be done on this result since the final value is very difficult to estimated and decide if it is accurate.

6   Monte Carlo simulations on the test instruments

This chapter presents how the test cases for the Asian and Barrier option were set up and the result from the simulations are analyzed. A comparison is done between the different techniques, evaluation how much time different simulation takes and subsequently a quick look on how the convergence for a European call option will be.

6.1   Test cases

Before the test case could be set up it was needed to decide some parameters; the time to maturity, the volatility, the strike price in relation to the current price, the interest rate and for the Barrier option the barriers. To decide these parameters experiments were done on the Asian option. First the time horizon was decided and the experiments started to compare the result when using constant time step or varying time step. The comparisons were done over the dimensions 130, 78, and 13, which mean that the varying time step was simulated over the days:

- 130, every day a week that is not a weekend, (Mo, Tue, We, Thu, Fri)
- 78, checking every Mo, We, Fri
- 13, checking every second Friday

In the case when constant time step is used, the time period from the start to the final date is calculated and divided by the dimensions. Figure 9 shows curves with varying time steps and constant time steps, and it can be seen that varying time step has another pattern. In the experiments only constant time steps will be used though varying time steps will give some random numbers more weight and meaning.
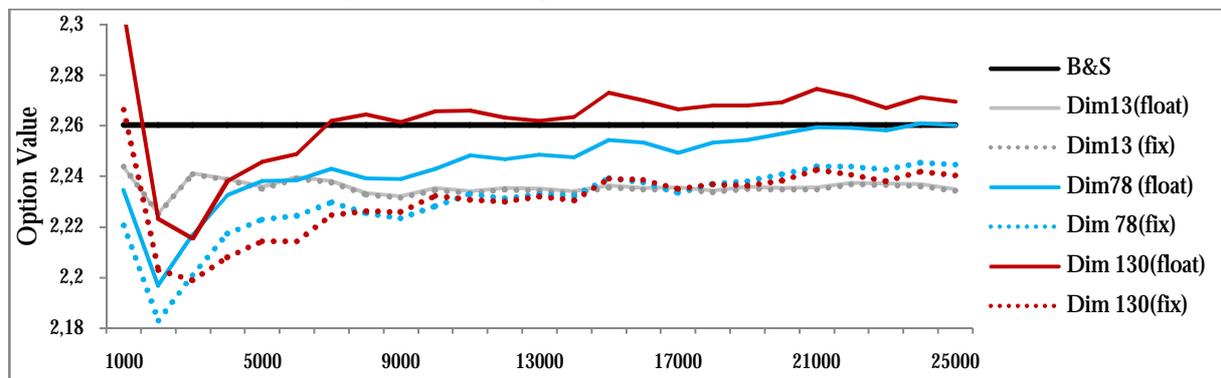

Figure 9. Sobol numbers, with constant time step (dotted lines) and varying time step (unbroken lines)

It is desirable to check the barrier as often as possible, but a higher dimension takes more time and of this reason a half year was decided as a time horizon. Some experiments were made to be sure that the time horizon does not affect the sequences properties too much.

The next parameter to decide was the interest rate, and the assumption that this parameter will not affect the curves was made and it was decided that the interest rate would be set to 5%, since it seems like a realistic interest rate. The current price was chosen to be constant and set to 100 SEK for all the test options.

The remaining parameters to examine were the strike price and the volatility. Decided were to value both in-the money options and out of the money options, with both high and low volatilities. The test cases that have been set up and their analytical values are represented in Table 8.

Option:	1	2	3	4	5	6
S_0	100	100	100	100	100	100
K	120	120	100	100	80	80
Vol	0.5	0.2	0.5	0.2	0.5	0.2
R	0.05	0.05	0.05	0.05	0.05	0.05
T	0.5	0.5	0.5	0.5	0.5	0.5
B(do)	75	75	75	75	75	75
B(uo)	125	125	125	125	125	125
B&S (A)	2.25	0.057	8.04	3.75	20.92	20.57
B&S (uo)	0.004	0.03	0.58	3.46	3.51	16.12
B&S (do)	7.94	1.02	14.34	6.89	23.74	22.15

Table 7. The values of the test options, where S_0 is the underlying value, K is the strike price, vol is the volatility, r is the interest rate, t is the time and B is the barriers. B&S is standing for the analytic value, (uo) stands for up-and-out and (do) for down-and-out

6.2 Simulation Results

After the test cases were set up the simulations started with the Asian option and then the two Barrier options. The simulations were done over the options in Table 7 with both Sobol and pseudo numbers. The methods were also combined with variance reduction techniques. The simulations were originally simulated over 1000-40000 numbers, over the dimensions 180, 150, 120, 90, 60, 30 and 10. In some cases to check divergent results or verify a statement, the options have also been simulated with other parameters.

6.2.1 Asian Call option

The simulation of the Asian Option started with the Sobol numbers on different start points; 256, 10000 and 20000, then the pseudo numbers have been initiated with different seeds to be able to investigate if there is a pattern.

The curves with Sobol numbers mostly converge for a value smaller than the analytic value, with exception the out of the money option with strike price 120SEK, where some dimensions seems to converge for the analytic value. Figure 10 shows an example of option number 3 for the dimensions 180 and 60 with different start points. Curves with a later start point seems to converge earlier and they also fluctuates less in the beginning of the sequence, after 5000 simulations the curves with start point 256 differs 3% from the analytic value, while the curves with a later start point is closer to 1%. But in the end the sequences converge for a similar value.
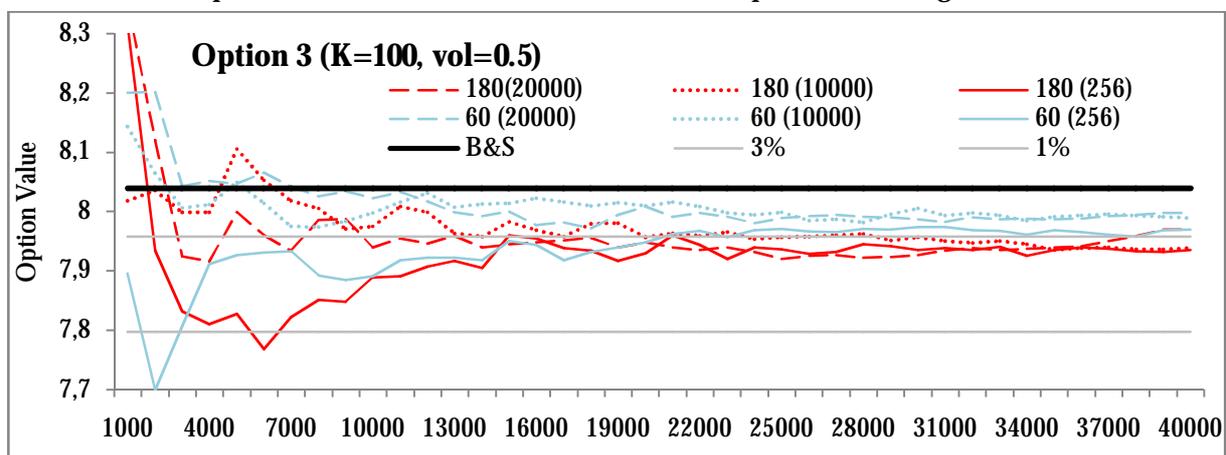

Figure 10. The curves over dimension 180 and 60 for option 3 simulated with Sobol numbers, the dashed line represents start point 20000, the dotted line; start point 10000 and the unbroken line; start point 256

For the pseudo numbers there is not this clear pattern, the curves fluctuate both over and under the analytic value and if changing the seed for the pseudo numbers the curves get a different look, see Figure 11. That means that the dimensions are not always converging for the same value. It seems like two of the sequences are converging quite the same but this is probably more a chance than a fact. The reason of this behaviour could be that the sequence is initiated with seeds giving values close to each other. For the pseudo sequence it seems like more simulations are needed to be sure of the convergence.
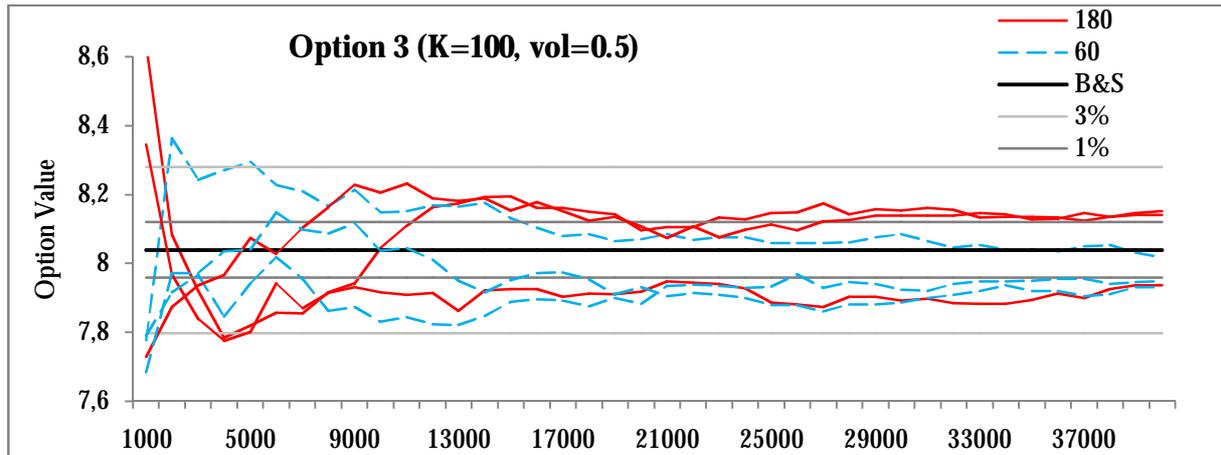

Figure 11. The curves over dimension 180 and 60, simulated with pseudo number and different seeds

The influence of the dimension

When all the simulations over the Asian option were done, the RMSRE and RMSE for both the pseudo and Sobol number were calculated. The diagrams for the RMSRE can be seen in Figure 12. For the RMSE the error for the pseudo numbers converges later than for the Sobol sequence, while the RMSRE shows a more similar convergence.
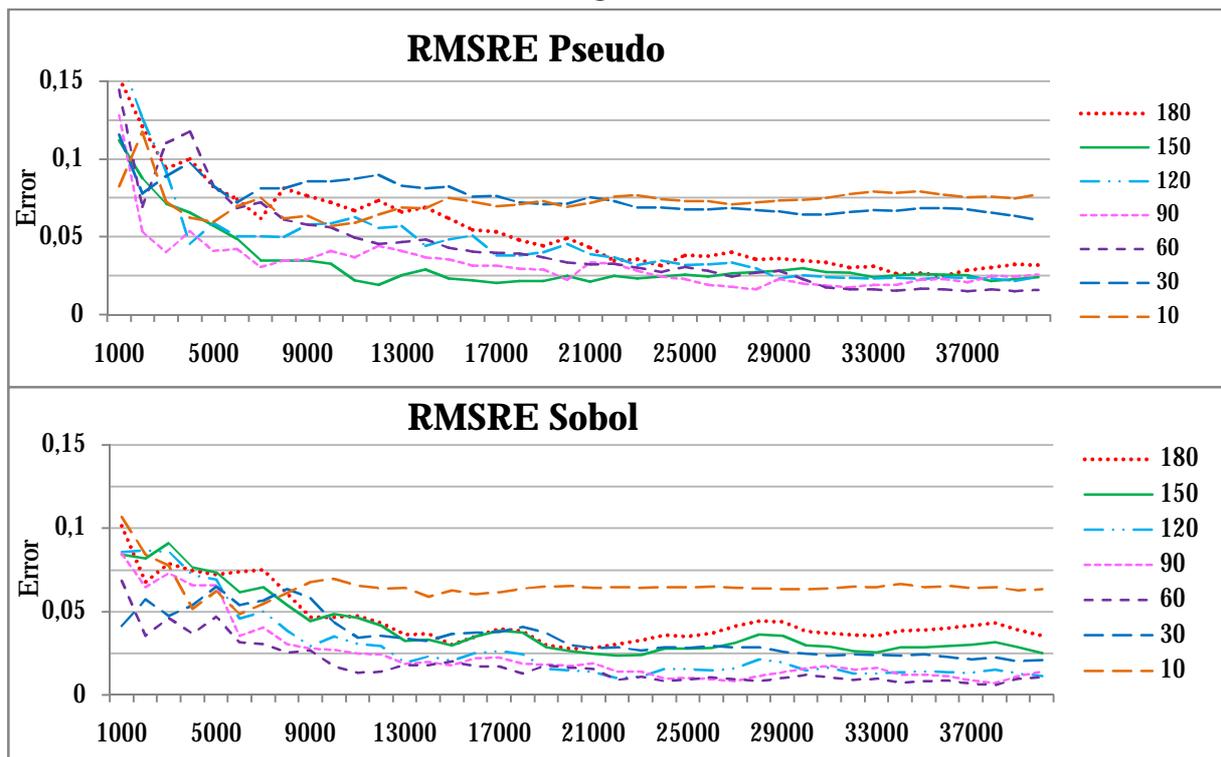

Figure 12. The RMSRE for the Sobol and pseudo sequences

The diagrams for the RMSE errors show that dimension 10 gives the biggest error compared to the analytic value for both the pseudo numbers and the Sobol numbers, and this is also for the RMSRE. For the Sobol sequence dimension 60 gives the smallest error (both RMSE and RMSRE) and for the pseudo sequence it is not as clear.

For the Sobol sequence when visually investigate the diagrams for every single option with different start points it seems like dimension 60 often gives the best result while the highest dimension; dimension 180, often is among the dimensions that perform worst. This assumption is confirmed from the RMSE and mostly from RMSRE where dimension 180 is the second worst after dimension 10.

It is more difficult to see what dimension that performs best and worst for the pseudo numbers since the curves fluctuates both above and under the analytic value. But the RMSRE shows that dimension 180 is not the dimension that converges best compared with the analytic value. If instead looking at the standard error to see if some dimension differs there is only a little difference between the different dimensions.

The assumption that as higher dimension as higher performance is not always true. An extra experiment was done on the dimensions; 60, 180 and 1000 and they were simulated over 1000-100000 samples. For the Sobol number dimensions 1000 converges best, thereafter dimension 60 and finally dimension 180, while for the pseudo numbers dimension 60 converges worse than the other two, but here the curves fluctuates a little bit more, see Figure 13.
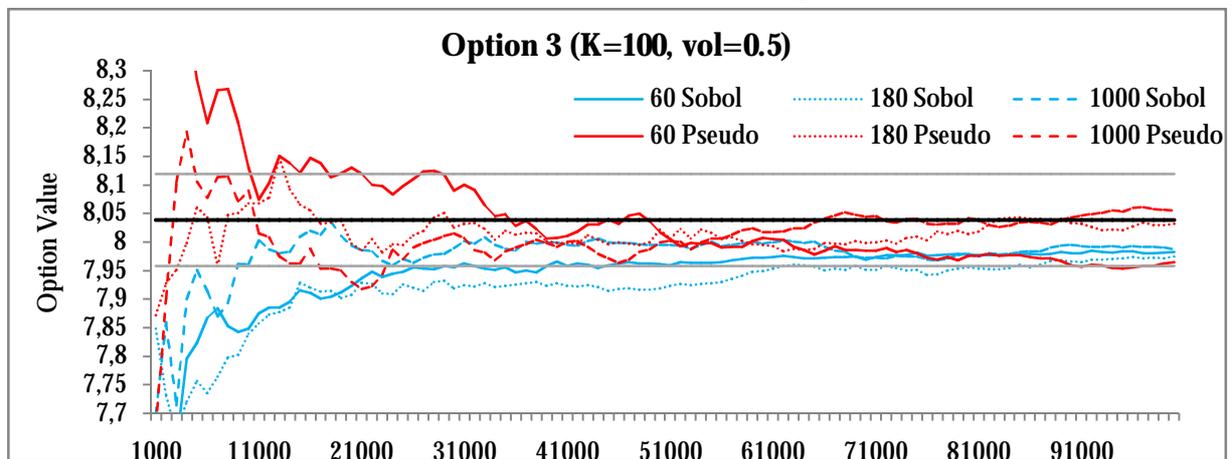

Figure 13. Option 3 simulated with 100 000 samples for dimension 60, 180 and 1000

Further investigations were done on the Sobol numbers and not on the pseudo number. This because it is easier to see a pattern on the Sobol numbers than on the pseudo numbers since they fluctuate more. The time parameter was in these experiments changed for one year. Interesting is that it does not seem like the time step have any influence of the look of the curve, rather it is the amount of random numbers. For example dimension 360 for a one year option give the same size on the time step like dimension 180 for a half year option. In the Figure 14 it can be seen that dimension 180 gives the same look on the curve in both the diagrams, and the same for dimension 30 and 60. The fact that dimension 10 differs most from the others is because it is only 10 random numbers that are used and not the fact that the time step is large.
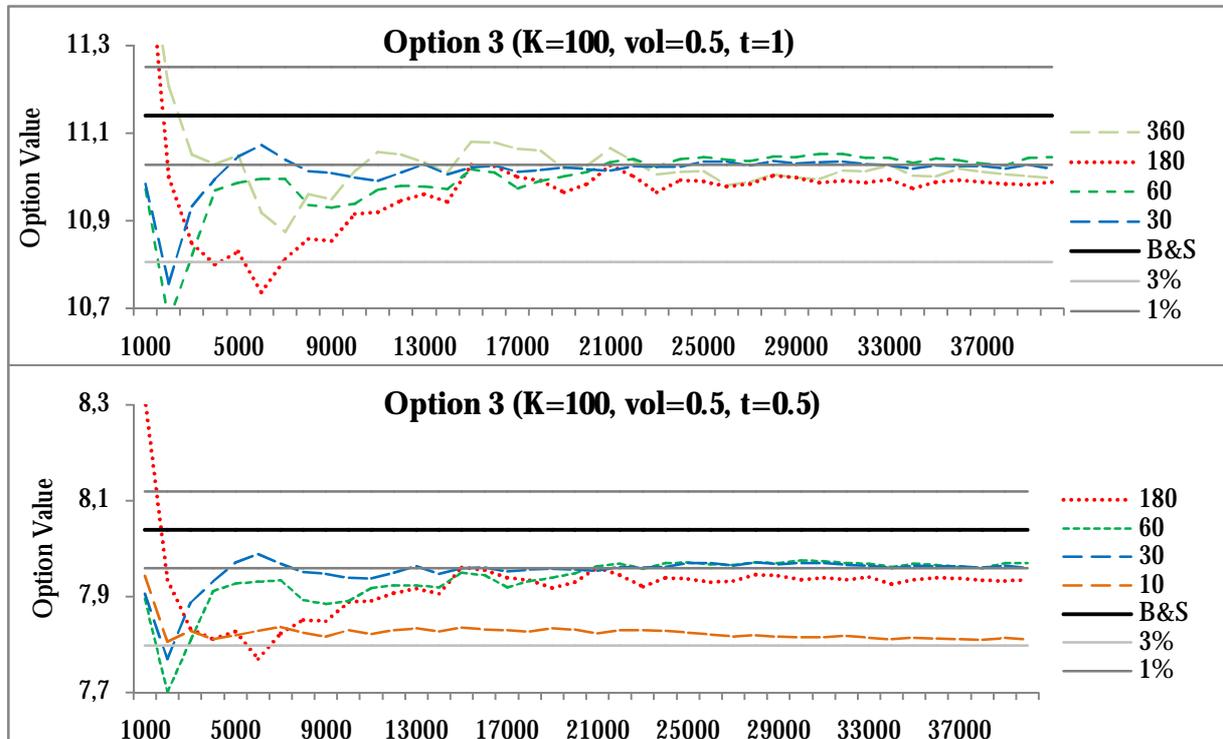
Figure 14. Option 3 simulated over a year and a half year

The margin and standard error

Comparing the options and the fluctuations of the curves, it can be seen that the value has a great affect on the fluctuations and the error margin. It seems like an option that has a high value has also a less error margin, but comparing to options close in price a high volatility also gives a higher error.

The options with a low strike price (option 5 and 6) are close in value (20.9 SEK versus 20.6 SEK). When looking to the fluctuations in percentage for the curves option 6 converges for a value that is 0.005 % while the curves for option 5 converges for a value that is 0.5% from the analytic value, and it also fluctuates more. Comparing option 3 and 4 where the strike price is the same as the underlying value, the option 3 is worth 8.0 SEK while option 4 is worth 3.75 SEK. The curves for option 3 and 4 look almost identical and the error is almost the same. The last comparison is done on option 1 and 2, these options fluctuate more in percentage than the others and it is a larger difference between the different dimensions. Option 2 is almost worthless and therefore the fluctuations in percentage are largest.

A high strike price seems to give more fluctuations and a lower value because an Asian call option with a high strike price, will lead to more worthless options. The payoff for a call option is $\max\{0, S_{avg} - K\}$ and in the case where the strike price is 120 SEK it is necessary that the average geometric value of the underlying asset is higher than 120 SEK to get an option of value. Having two options with different volatilities, it is more probable that the option with higher volatility reaches a higher value and therefore is worth more.

For the pseudo number it is easier to check the error by using the standard error. Here the result is the same as for the Sobol sequence; option 6 has the smallest error and option 2 has the greatest. The pseudo sequences are also compared with Sobol on the margin errors in the graph, and here the curves fluctuates more than for Sobol, especially for the options 5 and 6 that with Sobol numbers had a very small error margin.

Variance reduction techniques

The antithetic variate technique for the Sobol number gives no big difference on the performance. For the option 3, 4 and 5 the variance reduction technique is almost identical with the original Sobol numbers. For the option 6, the antithetic variate technique gives curves that are a little bit smoother, and for option 1 and 2 the antithetic variate reduces the value slightly.

When the antithetic variate technique is used for the pseudo numbers the standard error is reduced for all the options. The error is reduced mostly for option 6; here the standard error only becomes 5% of the original standard error. For option 1 the difference in the error is smallest, and the standard error is still 68% of the original standard error. See Figure 15.
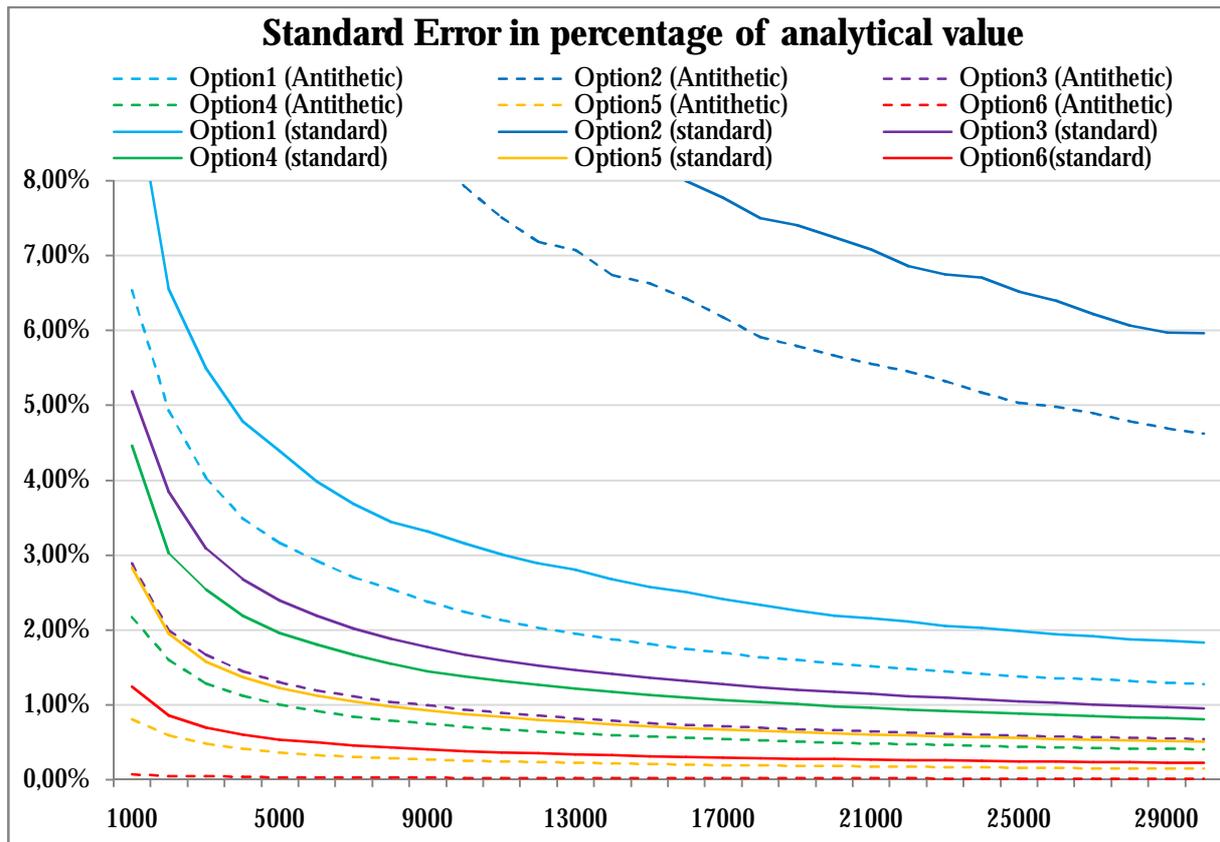

Figure 15. A comparison of the standard error in percentage of the analytic value for option 1-6, with standard pseudo numbers and antithetic variate technique

The control variate technique was also used, where the control variable was the geometric Asian option and the normal variable was the arithmetic Asian option. For the control variate technique there is no analytical value, but if comparing the fluctuations of the curve for the Sobol number, it is visible that the control variate technique has smaller fluctuations. The control variate technique also reduces the standard error for standard MC. Figure 16 presents the standard error in SEK.
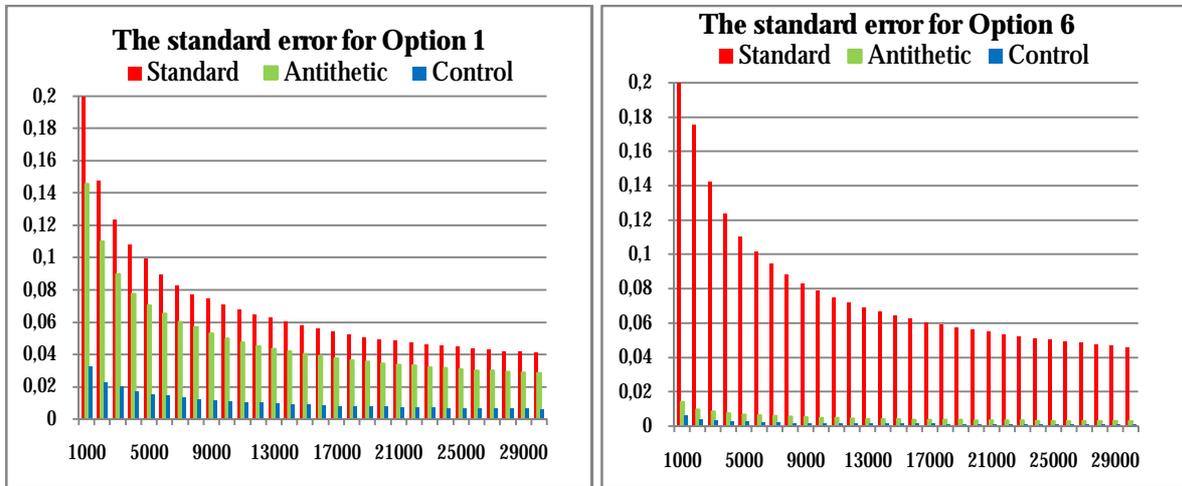
Figure 16. The standard error in SEK for option 1 and 6 with different variance reduction techniques

Time

The time to finish the execution for 40000 samples and the time for each sample can be seen in Table 8. Several comparisons are made. The time difference between Sobol and pseudo is very small. The time increases with 108% from standard pseudo to the antithetic variate technique. Comparing the difference between the dimensions it can be seen that if double the number of dimensions, the time increases around 96%.

Dimension	Sobol, 40000 samples	Sobol, 1 sample	Pseudo, 40000 samples	Pseudo, 1 sample	Antithetic, 40000 samples	Antithetic, 1 sample
180	101.50s	2.54 ms	106.28 s	2.66 ms	217.58 s	5.44 ms
150	85.31 s	2.13 ms	87.97 s	2.20 ms	184.17 s	4.60 ms
120	68.50 s	1.71 ms	71.50 s	1.79 ms	148.47 s	3.71 ms
90	51.05 s	1.28 ms	54.49 s	1.36 ms	113.42 s	2.84 ms
60	34.52 s	0.86 ms	36.14 s	0 .90 ms	75.56 s	1.89 ms
30	18.02 s	0.45 ms	18.61 s	0.47 ms	38.70 s	0.97 ms

Table 8. The time for each dimension for an Asian option

6.2.2   Up-and-out Barrier Call Option

For the Barrier option the simulations were first set up over the dimensions 180, 150, 120, 90, 60, 30 and 10 but later only over the dimension 180, 150, 120 since all the simulations show that a smaller dimension give worse values, for example in Figure 17 it is obvious that dimension 10 differs most, and also dimension 30 gives a worse result and these dimensions are therefore taken away in further analysis.
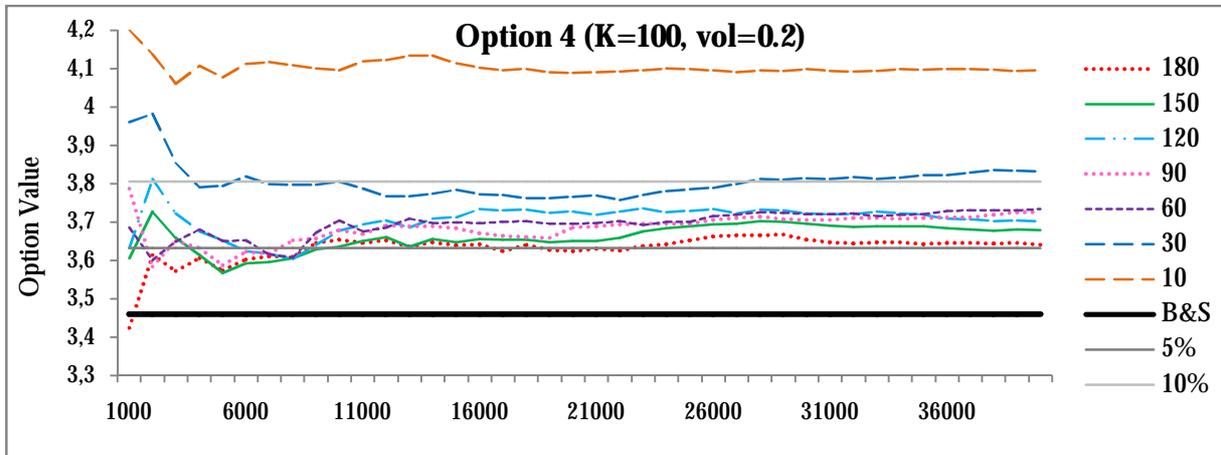
Figure 17. The up-and-out Barrier option (number 4), simulated with Sobol numbers

Figure 18 shows the curves for dimension 180 and 120 for option number 5 simulated with Sobol numbers, starting on point; 20000, 10000 and 256. It can be seen that the curves differ a little bit in converges with different start points especially for dimension 180. For this option it seems like the start point 20000 also starts to converge faster than the other sequences, this is not a common feature for all the options and dimensions. Comparing the maximum and minimum value for the three highest dimensions between 9000-19000 simulations leads to a result where the earliest start point 256 fluctuates more than the other two for all options.
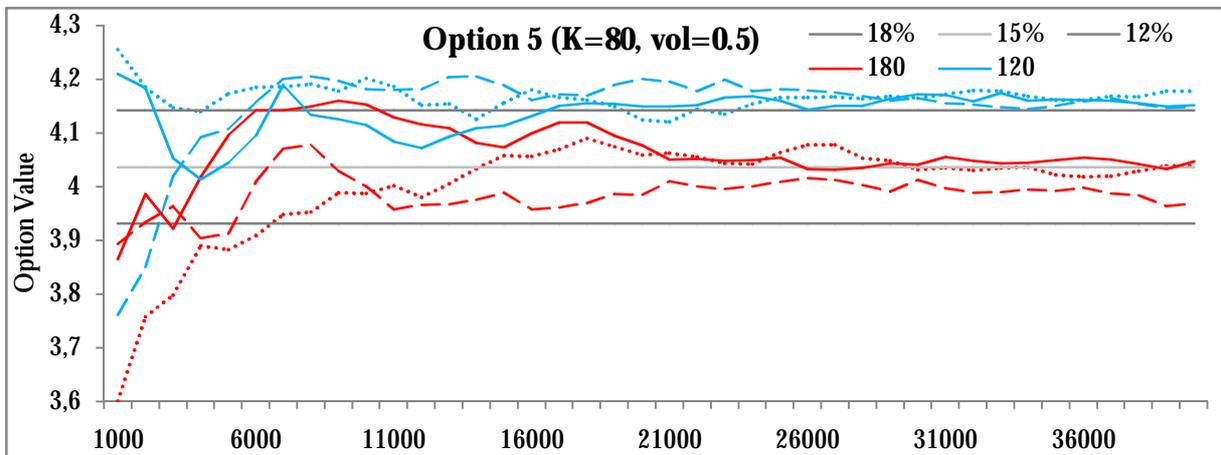

Figure 18. Sobol number with different start points (The dotted lines have start point 256, the unbroken lines have start point 10000 and the dashed lines have start number 20000)

The pseudo numbers were also simulated with different seeds and in difference with the Asian option where the different seeds gave completely different patterns, the up-and-out option give almost the same pattern no matter the seed. Another interesting phenomenon is that there is no big difference between the Sobol and pseudo sequences see Figure 19.
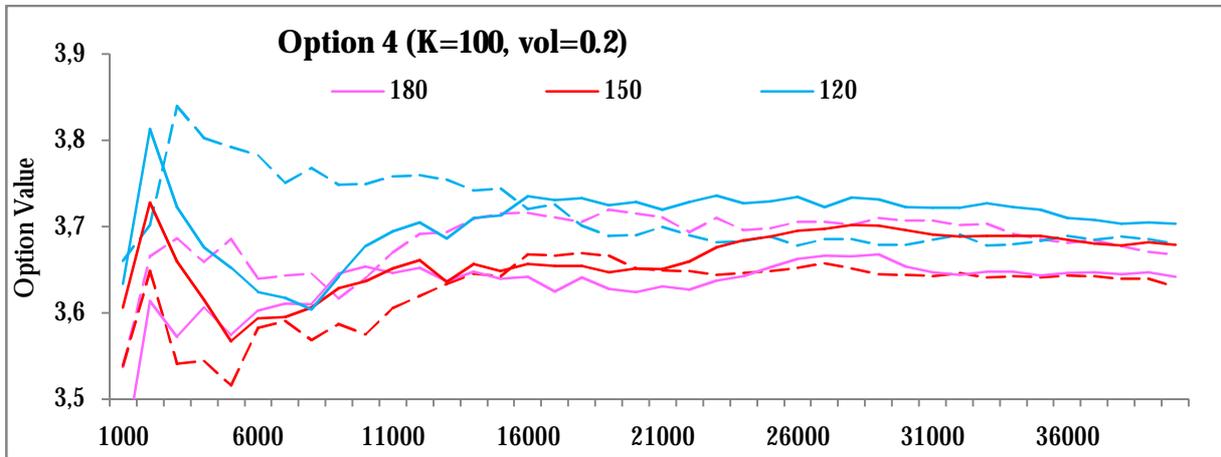
Figure 19. The difference between Sobol and pseudo, the dashed lines are pseudo numbers and the unbroken line represent the Sobol numbers

The influence of the dimensions

For the option in Figure 17 it can be seen the higher dimension the better performance. This is mostly the case for all the options simulated with Sobol numbers. For the pseudo number it is not always the dimension 180 converges best, it happens that dimension 150 is better but never lower dimensions. This can be due to the fact that more simulation should be done to get a better result.

That a higher dimension performs better than a lower shows Figure 20, (the option is simulated with Sobol numbers) where it is clearly that when using 1000 dimension the curve converges better for the analytical value. While dimension 180 converges for a value that is 15% bigger than the analytic value, dimension 1000 converges for a value that is 5% bigger than the analytic value. This result seems like a correct result, since the analytic value is calculated on a barrier that is checked continuously, and the more often the barrier is checked the higher probability it is that the underlying value reach the barrier, if only checking ten times per half year it is very likely that even if the barrier is reach during the time this moment is missed, and therefore a higher value on the option.
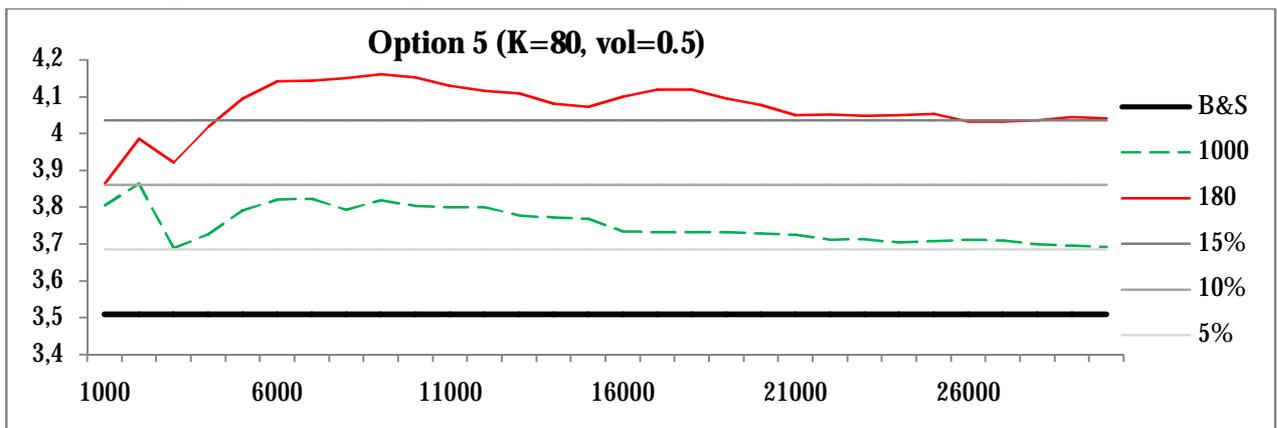

Figure 20. Sobol number dimension 180 (unbroken line) and dimension 1000 (dashed line)

Standard and Margin error

The standard error differs between the dimensions; dimension 10 has a larger standard error than dimension 180. But as for the Asian Option an option with a very low value has a larger standard error or margin error than the others. Option 1 is almost worthless and this can be explained because the option has a strike price on 120 SEK and a barrier on 125 SEK. As for the Asian option a high strike price, will lead to more worthless options because the payoff for a call option is $\max\{0, S_T - K\}$. In this case we also have a barrier so if the underlying value reaches a value

40

higher than 120 SEK it is also a high risk that the value increases over 125 SEK and the option becomes worthless. Since the option is worthless the margin error is large. If two options have a similar analytic value like option number 4 and 5, the option with largest volatility will have a larger standard error, the curves with larger volatility also fluctuates more.

The standard error in percentage is largest for option 1, followed by 2,3,5,4 and finally 6. The margin of the errors in the Sobol diagrams show that dimension 180 for option 1 converges for a value that is double size of the analytic value, while dimension 180 for option 6 differs less than 3%.

Antithetic variate technique

The antithetic variate technique for the Barrier option has not a significant difference for the Sobol numbers, the curves follow very much the same patterns; this can be seen in Figure 21. For the pseudo numbers the standard error is reduced. None of the options are reduced to the half error, but option 4 is closest where the error for dimension 180 is reduced with 45%, but for the rest of the options the standard errors with antithetic variate technique are about 70% of the original standard error.
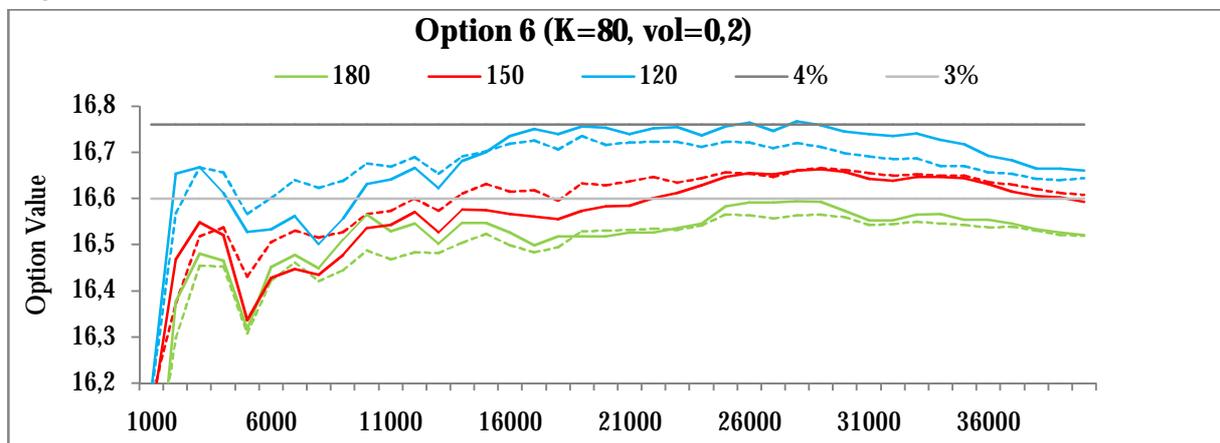

Figure 21. The difference between standard Sobol and antithetic variate technique, the dotted line represent the antithetic variate technique and the unbroken line standard Sobol.

Time

The parameter time will be analyzed in the next chapter 6.2.3, since the time is the same for the both barriers.

6.2.3   Down-and-out Barrier Call Option

The Sobol sequence for the down-and-out Barrier options shows a little unexpected behaviour. The analytic value is calculated on a Barrier option that is checked continuously; therefore it is possible to assume that the simulated curves should have a higher value exactly as for the up-and-out option. This is correctly for the lower dimensions but not the case for the higher dimensions that often converge for a value under the analytic value. Another assumption is the higher dimension the lower value (because the barrier is checked more often) but for the Sobol sequence the dimensions; 150, 120 and sometimes 90 have curves that converge for a value under the dimension 180. This is the appearance no matter start point.

These results were further analyzed and the options were simulated again with Sobol numbers but with the time to maturity as 1 year, and the dimension 360, 300, 240, 180, 120, 60 and 30. The three highest dimensions; 360, 300, and 240 follow the expected assumption, where the lowest dimension gives the highest value. When coming to dimension 120, the curves changes in result and dimension 120 converges for a lower value than several of the higher dimensions. Exactly as for the Asian option, the curves give roughly the same look for both a

year and a half year option, but the curves for the one year options converges for a larger value than the analytical value, see Figure 22.
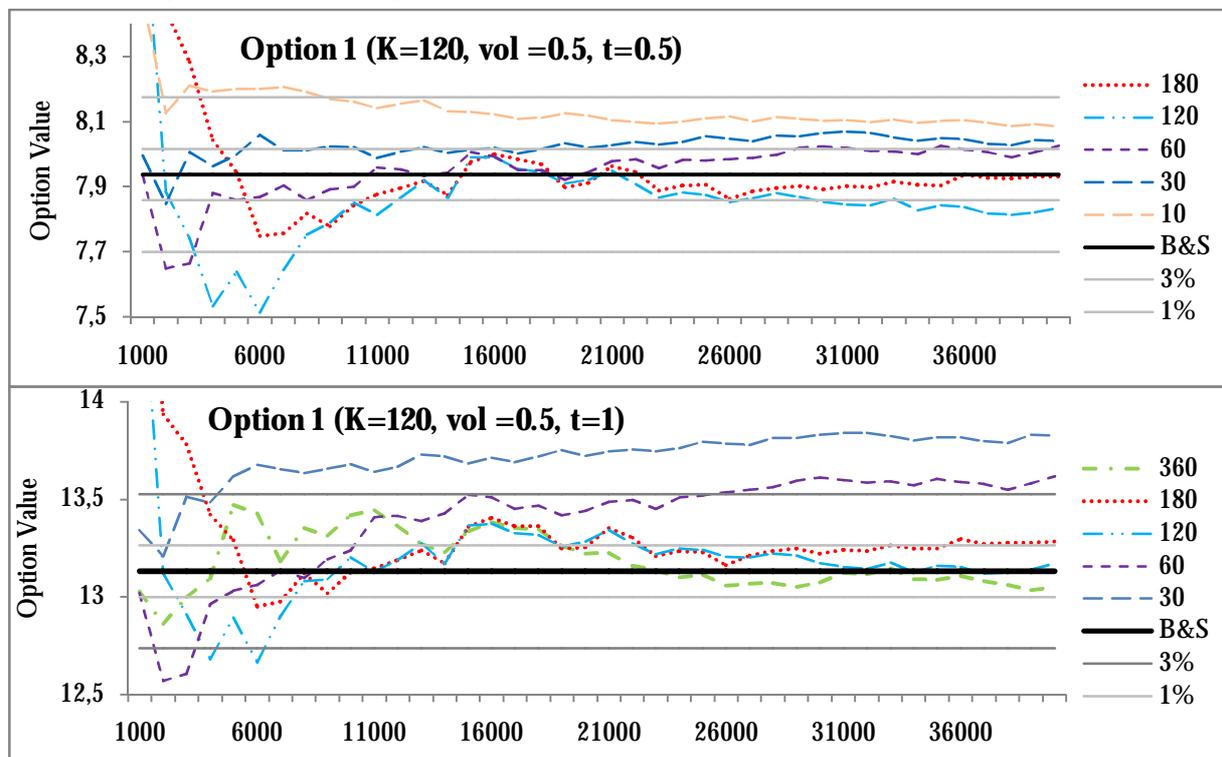

Figure 22. Option 1 simulated over a year and a half year with Sobol numbers

An explanation that several of the curves converge for a lower value than the analytical, may be that there are very few options that cease to exist because of the barrier. If only concentrating on the value of the down-and-in option (Table 9) it can be seen that it is almost worthless for the three options with low volatility and has a small value for the rest. The graphs approve that it is the options with small down-and-in values that converge for a lower value than the analytical.

t=0.5	1	2	3	4	5	6
C_{value} (b&s)	8.17	1.02	15.13	6.89	26.38	22.17
C_{value} (di)	0.24	0.00	0.79	0.00	2.64	0.02
C_{value} (do)	7.94	1.02	14.34	6.89	23.74	22.15
t=1						
C_{value} (b&s)	14.8	3.25	21.79	10.45	31.79	24.59
C_{value} (di)	1.65	0.00	3.22	0.01	6.40	0.24
C_{value} (do)	13.3	3.25	18.57	10.44	25.39	24.35

Table 9. The values for the down-and-out options

If instead inspecting the pseudo sequences the same appearance as for the Sobol is not observed. Here it seems to be a little bit more random, sometimes higher dimension converge for a higher value than the lower dimensions, and sometimes they converges for a value under the analytical. The curves also fluctuates a little bit more, but there is no big difference.

Increasing the number of samples to 100 000 and sample option 3 for a half year with 1000 dimension both for the Sobol and pseudo numbers gives the result in Figure 23. In the diagrams it can be seen that if just inspecting the samples up to 40000 samples the dimension 1000 shows a higher converges than dimension 180, and this is also true for all the options 1-6, but with more

samples they converge for the same number. The pseudo numbers fluctuates more, but in the end they converge for a higher value than Sobol.
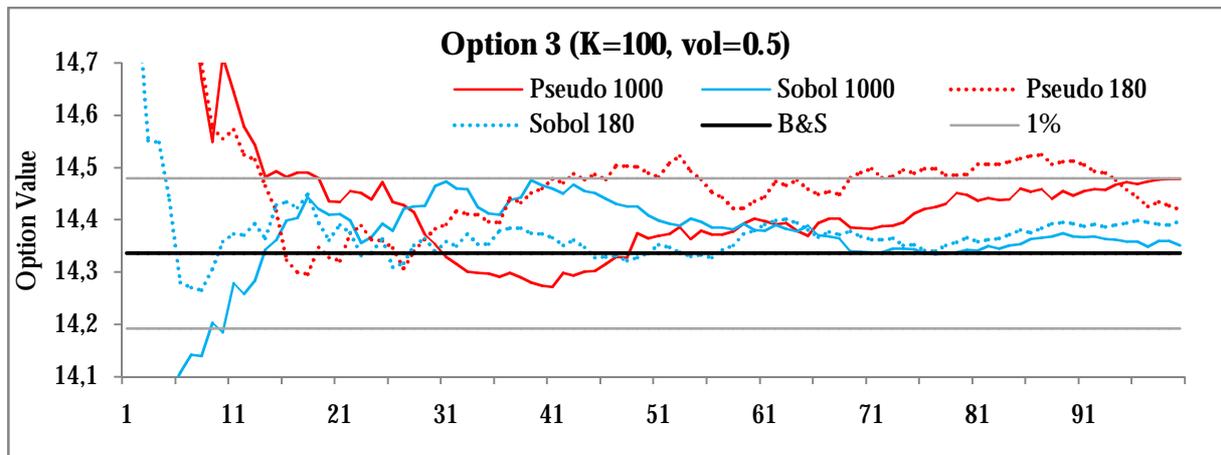

Figure 23. The pseudo and Sobol numbers simulated with 10000 samples for dimension 180 and 1000

The assumption from previous cases that an option with a very low value has a larger standard error, and to option with a high value implies that a higher volatility gives a higher standard error, is approved here too. All the down-and-out options have a higher value than in previous cases, and it can also be seen on the fluctuations of the curves, because they fluctuates less in value. Most of the curves converge for a value that differs less than 1% from the analytic value. If looking to the standard error option 2 has the largest error and option 6 the smallest.

As in previous cases the antithetic variate technique has no improvement on the Sobol sequences, but improves the pseudo sequences. Mostly option 6 is improved, that now only correspond to 16% of the original standard error, while dimension 2 has the worst improvement and the new standard error comes up to 70% of the original.

Time
The time to finish the execution for 40000 samples and the time for each sample can be seen in Table 10. Several comparisons are made; the difference between Sobol and pseudo is very small, in the case with 100000 samples for 1000 dimensions, the pseudo takes 41 minutes and 13 seconds, while Sobol takes 38 minutes and 16 seconds. The time increases with 113% from standard pseudo to the antithetic variate technique and finally if comparing the difference between the dimensions it can be seen that if double the number of dimensions, the time increases around 94% no matter technique.

Dimension	Sobol, 40000 samples	Sobol, 1 sample	Pseudo, 40000 samples	Pseudo, 1 sample	Antithetic, 40000 samples	Antithetic, 1 sample
180	82.27 s	2.06 ms	87.58 s	2.19 ms	184.99 s	4.62 ms
150	68.30 s	1.71 ms	72.49 s	1.81 ms	154.16 s	3.85 ms
120	54.74 s	1.37 ms	58.81 s	1.47 ms	124.39 s	3.11 ms
90	42.13 s	1.05 ms	44.13 s	1.10 ms	95.51 s	2.39 ms
60	29.17 s	0.73 ms	29.75 s	0.74 ms	63.77 s	1.59 ms
30	15.36 s	0.38 ms	15.83 s	0.40 ms	33.06 s	0.83 ms

Table 10. The time for the Barrier options

6.3 Comparison between the different techniques

Finally some more test cases were set up on random instruments where the dimension and the number of samples were changed, to see what affected the time, and to investigate the convergence when only having one dimension.

6.3.1 Time

If setting up a basket option with three correlated assets, with the number of sample 10000 and dimension is 500, the time for the different simulations are:

- Sobol takes 5 minutes and 25 seconds
- Pseudo takes 5 minutes and 34 seconds
- Pseudo with antithetic variate technique takes 11 minutes and 50 seconds

When the parameters are changed to 30000 number of samples and the dimension to 50, the time for the different simulations are:

- Sobol takes 1 minutes and 40 seconds
- Pseudo takes 1 minutes and 46 seconds
- Pseudo with antithetic variate technique takes 3 minutes and 46 seconds

1 million simulations on an asset of 1 dimension give the time:

- Sobol takes 74 seconds
- Pseudo takes 82 seconds
- Pseudo with antithetic variate technique takes 115 seconds

This final result is compared with a simulation where 500 dimensions are used but only 2000 samples, that means that the path generator is called totally 1 000 000 times.

- Sobol takes 22 seconds
- Pseudo takes 23 seconds
- Pseudo with antithetic variate technique takes 47 seconds

From this simulations it can bee seen that if few dimensions are used the antithetic variate technique does not take the double time compared to standard, while for more dimension it takes often more than the double time. It can also be seen that the part that affects the time the most is the number of samples and not the dimensions.

6.3.2 Convergence European Call Option

Figure 24 shows two diagrams over an European call option.
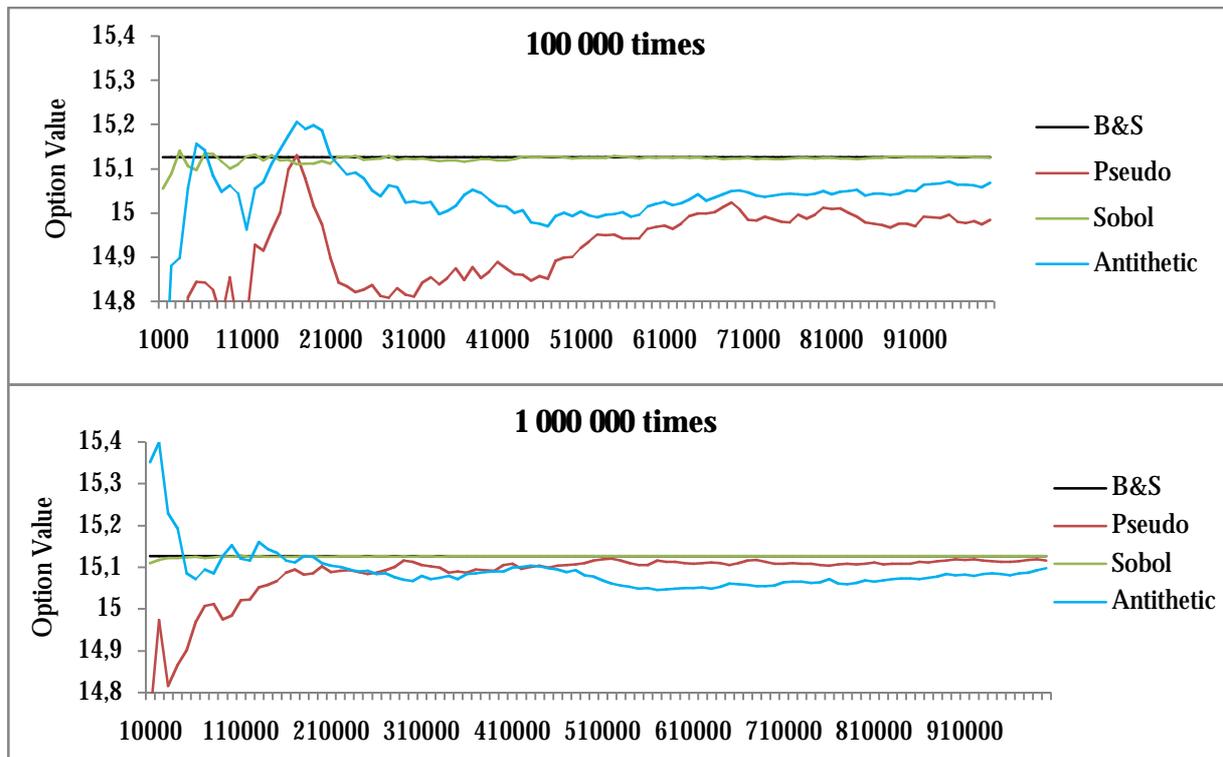
Figure 24. The convergence of a European call option when simulating 100 000 samples and 1 000 000 samples

The diagrams show that the Sobol numbers converge must faster than the pseudo random, even when variance reduction techniques are used. The variance reduction technique reduces the standard error compared with the pseudo numbers, the new standard error is around 58% of the original. But in the diagram for 1 000 000 the standard pseudo converges better.

7   Conclusions and Recommendations

This chapter presents the conclusions and recommendations of the experiments and analysis.

7.1   The different MC methods

Considering the different QMC methods the theory points out that the Halton sequence is not a *(t,s)-sequence* or a *(t,m,s)-net*, and it is also obvious that the sequence suffers from high correlation already in early dimensions, therefore this sequence is not recommended to be implemented in Quantlab. The Faure sequence also seems to perform worse than Sobol, but this has not been examined enough to draw a certain conclusion. The Faure sequence can be interesting in the future, but it must then be a part of the function library of Quantlab like the Sobol sequence.

It is recommended to implement the variance reduction techniques in the framework, since they reduce the error. But for a general framework the control variate technique cannot be implemented in the framework, it requires an analytic value and this should be very difficult to solve. The user must implement the technique himself for example in the payoff function. This can be done if it is assumed that the control variate technique uses the same vector with the simulated underlying values, to calculate both the instrument and the control variable or if the user implement the control variable and standard variable as two assets and use the same random numbers when simulating them

There are several more variance reduction techniques that could be interesting, like Brownian Bridge, Randomized QMC, and the more complicated Stratified Sampling and Importance Sampling. These methods are not treated by the Master's thesis and further analysis is needed before drawing a conclusion. Especially randomized QMC would be interesting to investigate further, since it is a method that combines the best features from QMC with the best features from ordinary MC and makes it possible to measure the error of the simulation. If Algorithmica wishes to implement a MC generator it would be interesting to investigate these methods further.

7.2   The general framework

The chosen solution in this Master's thesis is to design a general framework by using function pointers. This is not a problem-free solution and several conflicts and issues are defined in the analysis of the function pointers in chapter 5.3. A disadvantage is that the developed framework is very general and should apply on specific instrument. If writing a specific code it would probable reduce the computer complexity

The chosen ways to generate the volatility and interest rate are not optimal; as long as the parameters are constant it is easy to just return the previous parameter. But parameters are non constant they can be calculated in several ways. For example a non constant volatility can be expected to follow a Wiener process, this Wiener process is often independently of the Wiener process the assets is assumed to follow. This means that the random numbers for the volatility and the path should be generated independently, which is not the case in this framework.

A potential conflict arises when an instrument is valued by Sobol numbers, where the valuation is affected by the dimensions. The dimensions are for a basket option the number of stocks in the basket, and for a Barrier option the dimensions are the number of time step. It is difficult to combine these two definitions of the dimension, and since there may be a basket that exists of path-dependent instrument, no respect is taken to the fact that the dimension also can be the number of assets.

There are several potential problems that are identified in this thesis, for example if non constant parameter follows a Wiener process that is independent from the path or if the path follows a stochastic process where the probability functions is not lognormal. Another aspect of the path generator is that it cannot be guaranteed with the implemented proposed framework that the input values chosen for the function pointer are enough for generating all kinds of paths.

For this a more thorough analysis is required and how to incorporate the other stochastic processes than GBM. In this solution the path must be lognormal, since the random numbers are normally distributed; if another probability function is chosen, the way to generate random numbers must also be changed.

As it is now, the general framework requires a lot from the user. The user must know how the antithetic variate technique is implemented, remember to discount the value when calculating the payoff, keep in mind how the stop criterion matrix is built up, and have knowledge how to generate the path etcetera. The question one can ask is; if the user can implement all the function pointers himself, is he/she not good enough to implement the whole MC simulation? A general framework would reduce the usability and the program is more difficult to use than a more specific program.

A solution to overcome the potential sources of conflicts would be to only implement a framework that handles simple path-dependent options where the underlying is following a GBM and the parameters are constant. The chosen test instrument; the Asian and the Barrier option can easily be valued in the framework. My conclusion is that it is difficult to create a framework that is as general as Algorithmica desires and still is easy to use.

7.3   The performance of the different techniques

On the question on which method that is the best to use for the chosen test instruments, there is no clear answer. There are advantages and disadvantages with all the techniques. From further analyses it can be seen that for the Sobol numbers, it is not the size of the time step that affects the performance of the curves but rather the dimension. No matter if the samples are over a half year or a year, the dimension 180 gives the same result and convergence. It is not always that a higher dimension gives a better result, and the difference is very small. Only for the Up-and-out Barrier option it is a significant difference when increasing the dimensions. Therefore it is not always recommended to choose as high dimension as possible, since the performance time increases a lot. The same conclusion obtains for the pseudo sequences, but here the behaviour is more fluctuating.

The start point or the seed of the sequence seems to affect the Sobol numbers where often a higher start points seems to give a faster convergence, but the difference is small and not constant. Previous research indicates that the sequence is more uniform in early samples but when the start point is zero the sequence produces the same number for all dimensions; therefore the proposal is that the start point should not be zero, but neither too high. For the pseudo random numbers the seed does not seem to affect the convergence.

The more samples drawn the better the convergence, but more samples also take more time. Here the standard error is a very good measure to know when to finish the simulation, but if Sobol numbers is used it is not possible to measure the standard error. What seems to be the main advantage with the Sobol sequence is that the curves often fluctuate less and tend to converge earlier than with the pseudo numbers. For the Up-and-out Barrier option the difference between them is very small and in that case, pseudo numbers are recommended since the standard error can be measured.

The variance reduction techniques reduce the standard error. An advantage with the control variate technique as it is implemented now is that it does not increase the time compared with standard MC, because the same sequence is used twice. But the technique requires that the user finds a good control variable. The antithetic variate technique reduces the standard error, but in the other hand it takes more than the double time. The reduction of the error is very different sometimes the new standard error is only some percent of the original one, while other times it is as much as 70% of the original error. Therefore it is difficult to always recommend this solution. It seems like for options with high values and low volatilities the standard error is reduced the most.

The conclusion is that the Sobol numbers often seems to converge better and gives less fluctuation than the pseudo numbers. Other research supports this observation. Glasserman points out something important when answer the question what technique that is recommended to be used (Glasserman, p. 276): "*The choice of technique should depend on the available information and on the time to tailor a general technique to a particular problem. An understanding of the strengths and weaknesses of alternative methods and familiarity with examples of effective applications are useful in choosing a technique.*"

From this conclusion it follows that it is important that the user has an understanding of the instrument that should be valued, the stochastic process it follows and the advantages and disadvantages of different MC methods.

A way to choose a method is to do several experiments. If a satisfactory algorithm is found for one instrument, the algorithm could probable also apply on similar problems. Glasserman suggest that starting with standard MC, and thereafter use variance reduction techniques and continue with an investigation if QMC increases the accuracy further. (Glasserman, p. 336)

7.4 Recommendations

From the conclusions follows that it is not recommended to implement the general framework at it is designed today. This solution leads to too many conflicts and the usability is low.

Concerning the different MC simulation methods a recommendation on what method to use, which dimension and the number of trials, cannot be given, though the result is varying. The recommendation is that when using MC simulation experiments should be done.

References

Antonov, I. A., & Saleev, V. M. (1979). An economic method of computing LP-sequences. *USSR Journal of Computational Mathematics and Mathematical Physics*, 19:252-256.

Black, F., & Scholes, M. (1973). The pricing of Options and Liabilities. *The Journal of Political Economy*, 637-654.

Boyle, P., & Tan, S. K. (1997). Quasi Monte Carlo Methods. *AFIR conference in Cairns*. Cairns Australia.

Bratley, P., & Fox, B. L. (1988, March). ALGORITHM 659 Implementing Sobol's Quasirandom Sequence Generator. *ACM Transactions on Mathematical Software*, pp. 88-100.

Bromley, B. C. (1996). Quasirandom Number Generators for Parallel Monte Carlo Algorithms. *Journal of Parallel and Distributed Computing*, 38:101-104.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. New York: Springer.

Global Derivatives. (n.d.). *Global Derivatives: Options Database: Barrier Options*. Retrieved November 17, 2008, from Barrier Options: http://www.global-derivatives.com/index.php?option=com_content&task=view&id=13&Itemid=31

Haugh, M. (2004). *IEOR E4703: Monte Carlo Simulation*. Retrieved 09 25, 2008, from The Monte Carlo Framework, Examples from finance and generating correlated random variables: http://www.columbia.edu/~mh2078/MCS04/MCS_framework_FEegs.pdf

Hull, J. C. (2006). *Option, Futures and Derivatives*. New Jersey: Prentice Hall.

Hull, J., & White, A. (1998). The Use of the Control Variate Technique in Option Pricing. *Journal of Financial and Quantitative Analysis, 23*, 237-251.

Joe, S., & Kyo, F. Y. (2008). *Sobol Sequence Generator*. Retrieved 10 14, 2008, from Frances Kuo Homepage: Sobol Sequence Generator: http://web.maths.unsw.edu.au/~fkuo/sobol/

JP Morgan Chase & Co. (2008). *Prospekt Obligation Smart Start 2 - Överkurs (Eng).* Retrieved Oktober 13, 2008, from Kaupthing Bank: Strukturerade produkter: Prospekt: http://www.kaupthing.se/Hem/Private-Banking/Vart-senaste-erbjudande/Prospekt

Lekvall, P., & Wahlbin, C. (2001). *Information för marknadsföringsbeslut*. Göteborg: IHM Förlag.

Li, J. X., & Mullen, G. L. (2000). Parallel Computing of a Quasi-Monte Carlo Algorithm for Valuing Derivatives. *Journal of Parallel Computing*, 641-653.

McLeish, D. L. (2005). *Monte Carlo Simulation and Finance*. New Jersey: John Wiley & Sons.

MCNeill, P., & Chapman, S. (2005). *Reserach methods*. Oxon: Routledge.

Mullen, G., Mahalanabis, A., & Niederreiter, H. (1995). Tables of (t,m,s) - nets and (t,s) sequence paramters. In H. Niederreiter, & P. Shiue, *Monte Carlo and Quasi Monte Carlo Methods in Scientific Computing* (pp. 58-86). New York: Springer.

Paskov, S. H. (1994). *Computing High Dimensional Integrals with Application to Finance*. New York: Department of Computer Science, Colombia University.

Zhang, P. G. (1997). *Exotic Options*. Singapore: World Scentific.

Appendix I

This appendix holds on description of the test instrument Smart Start.

Smart Start

Smart Start is a derivative issued by JPMorgan Chase & Co (Prospekt Obligation Smart Start 2 - Överkurs (Eng)). It is an index linked note that forms a basket of the equally weighted indices:

- Dow Jones STOXX Select Dividend® 30 (EUR) Index
- Dow Jones Asia/Pacific Select Dividend® 30 (EUR) Index
- Dow Jones U.S. Select Dividend Price Return Index

The subscription period is from 1 September 2008 to 3 October 2008. The initial valuating date is the 15 October 2008. The nominal amount is SEK 10000per note, and the minimum trading size is one note, thereafter the notes can only be traded in multiples of one note.

Smart Start has an issue date the 23 October 2008, and the issue price is 112 percent of the aggregated normal value.

The maturity date is the 23 October 2013 and it is also the redemption date (subject to earlier redemption in accordance to General Note Conditions).

The redemption amount is index linked and at maturity date the holder will receive an amount in SEK. The amount is calculated with the formula:

$$SEK\ 10000 * [1 + (P * Option\ Payout)]$$

P means the participation, an amount to be determined by the issuer on or around the 14 October 2008, and the amount shall not be less then 1.35. Option Payout is an amount calculated by the formula:

$$max\left(0; \frac{Final\ Basket\ Differential}{Lowest\ Basket\ Differential} - Strike\right)$$

Final Basket Differential is the arithmetic mean of the Asiatic end of each index. The Asiatic end is the arithmetic average over the closing index level at 13 dates, with start from 15 October 2012 and then taking the 15th day on each calendar month to the 15th September 2013, the final average date is the 7 October 2013. In each case adjustment for non scheduling trading dates shall be done.

Smart Start has a look-back start. The Lowest Basket Differential is the lowest arithmetic mean of the quotient of the value at the valuation day and the initial value. The valuation days, ten in total:

- The initial valuation date
- 22 October 2008
- 29 October 2008
- 5 November 2008
- 12 November 2008
- 19 November 2008
- 26 November 2008
- 3 December 2008
- 10 December 2008
- 17 December 2008

Strike is one that is 100 percent.

Appendix II

This appendix describes the principles behind risk free pricing and the development of Black and Scholes Formula.

Risk-free pricing

One of the fundamental concepts concerning valuation of financial derivatives is that it must be done in a risk-free world (Q) (Hull J. C., pp. 590-596). Because most assets are risky it is necessary to change the probability measure to the risk neutral measure. An important concept is martingales. A martingale is a Wiener process with a zero drift, which implies that the expected value is equal to the present value.

The market price of risk measures the trade-off between risk and return. In the traditional risk-free world, the market price of risk is zero, and the expected return μ is equal to the risk free rate r. The market price of risk can be defined as:

$$\lambda = \frac{\mu - r}{\sigma}$$

Instead of measuring the derivative in units of a currency, it can be measured in units of another derivative, called numeraire. If valuing a derivative X by using the numeraire Y, the relative price of X in Y units will be a martingale for all X if the market price of risk is chosen to be the volatility of the numeraire Y. Then it can be showed that the connection between X and Y is:

$$x_0 = y_0 E_y \left[\frac{x_T}{y_T} \right]$$

This result is called the equivalent martingale measure.

A common choice of numeraire is a money market account that is a risk less security; worth $1at time zero and then grows at the risk-free rate r. The value of a derivative with a money market account as a numeraire becomes:

$$V(0) = e^{-rT} E^Q[V(t)]$$

The Black and Scholes Formula

Fischer Black, Myron Scholes and Robert Merton have developed a very famous model for pricing option. According to Black and Scholes (pp. 640-641) it is necessary to assume some ideal condition to derive their formula:

- The stock price is following a GBM
- The short-term interest rate is known and constant
- There are no dividends
- There are no transaction costs
- Short selling of securities are permitted
- The derivative is European
- The valuation is done in a risk-free world

Under these conditions Hull (p. 291) shows that it is possible to create a portfolio of a stock (S) following a GBM:

$$dS = \mu S dt + \sigma S dZ \ (A.1)$$

and a European derivative (f). The value of f is a function of S and t, and by using Itô:s Lemma the fucntion may be written as:

$$dG = \left(\frac{\partial G}{\partial S}\mu S + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}\sigma^2 S^2\right)dt + \frac{\partial G}{\partial S}\sigma S dZ \quad (A.2)$$

The portfolio created will consist of a short position of one derivative and a long position of $\partial f/\partial S$ stocks:

$$\Pi = -f + \frac{\partial f}{\partial S}S \quad (A.3)$$

By using the time discrete versions of equation (A.1), (A.2) and (A.3), and substituting (A.1) and (A.2) into (A.3) we get the equation:

$$\Delta\Pi = \left(-\frac{\partial f}{\partial t} - \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)\Delta t \quad (A.4)$$

The elimination of the wiener processes in (A.1) and (A.2) can be done since both the derivative and the stock follow the same wiener process.

If the position of the derivative is adjusted continuously; the risk of the hedge position is zero a very short time period. The expected return of the portfolio must then be the risk-free interest rate otherwise there is arbitrage opportunities. The change in a portfolio during a small time interval is $\Delta\Pi = \Pi r \Delta t$. By replacing the $\Delta\Pi$ and Π with equation (A.3) and (A.4):

$$\left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)\Delta t = r\left(f - \frac{\partial f}{\partial S}S\right)\Delta t \Longrightarrow$$

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2 = rf$$

This result is called the Black-Scholes-Merton's (B-S-M) differential equation ant it must be satisfied for all derivatives where the underlying value follows a GBM, otherwise there exist arbitrage opportunities:

From the B-S-M partial equation and together with boundary condition for a European call option, Black and Scholes (p. 644) have developed the Black-Scholes formula. With the money market account as a numeraire the value of a call option whose payoff is $\max\{0, S_T - K\}$ becomes:

$$c = e^{-rT}E^Q[(S(T) - K)]$$

where the stock price follows a GBM. Since S has a lognormal distribution, the call price can be written as:

$$c = S_0\phi(d_1) - Ke^{-rT}\phi(d_2)$$

where

$$d_1 = \frac{\ln(S_0/K) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

From this equation several modifications can be made. If the underlying value of a derivative is a stock with a continuous dividend q the rate of return in a risk-free world is $\mu = r - q$ which implies that the stock will follow the process:

$$dS = (r - q)S dt + \sigma S dz$$

This equation is also the equation used for a stock index. A stock index is compounded by several different stocks often with dividend and all the dividend rates together can be seen as a continuous dividend for the whole stock index.

There exist an analytic solution for an Asian option with geometric average, if having a call option the Black & Scholes's formula will look like:

$$c = S_0 e^{-bT}\phi(d_1) - Ke^{-rT}\phi(d_2)$$

where

$$d_1 = \frac{\ln(S_0/K) + \left((r-b) + \frac{1}{2}\sigma_A^2\right)T}{\sigma_A\sqrt{T}}$$

$$d_2 = d_1 - \sigma_A\sqrt{T}$$

and b is the dividend yield:

$$b = \frac{1}{2}\left(r + q + \frac{\sigma^2}{6}\right)$$

and the volatility σ_A:

$$\sigma_A = \frac{\sigma}{\sqrt{3}}$$

For a Barrier option where the barrier is checked continuously there also exist an analytical solution. Reiner and Rubenstein provide a formula for all the eight types of Barrier options. (Global Derivatives). If the barrier is greater than the strike price for the up-and-out call the value is calculated as:

$$c_{up-out} = c_{B\&S} - c_{up-in}$$

where B is the barrier, K is the strike price and B&S is the Black & Scholes original formula, the c_{up-in} is calculated (Hull J. C., p. 534):

$$c_{up-in} = S_0 e^{-qT}\phi(x_1) - Ke^{-rT}\phi\left(x_1 - \sigma\sqrt{T}\right) - S_0 e^{-qT}(B/S_0)^{2\lambda}[\phi(-y) - \phi(-y_1)]$$
$$+ Ke^{-rT}(B/S_0)^{2\lambda-2}\left[\phi\left(-y + \sigma\sqrt{T}\right) - \phi\left(-y_1 + \sigma\sqrt{T}\right)\right]$$

Where:

$$\lambda = \frac{r - q + \sigma^2/2}{\sigma^2}$$

$$y = \frac{\ln\left(B^2/S_0 K\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

$$y_1 = \frac{\ln\left(B/S_0\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

$$x_1 = \frac{\ln\left(S_0/B\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

For a down and out call option where the barrier is less or equal to the strike price the procedure resembles of the up and out option:

$$c_{down-out} = c_{B\&S} - c_{down-in}$$

Where

$$c_{down-in} = S_0 e^{-qT}(B/S_0)^{2\lambda}[\phi(y)] - Ke^{-rT}(B/S_0)^{2\lambda-2}\left[\phi\left(y - \sigma\sqrt{T}\right)\right]$$