UPPSALA
UNIVERSITET

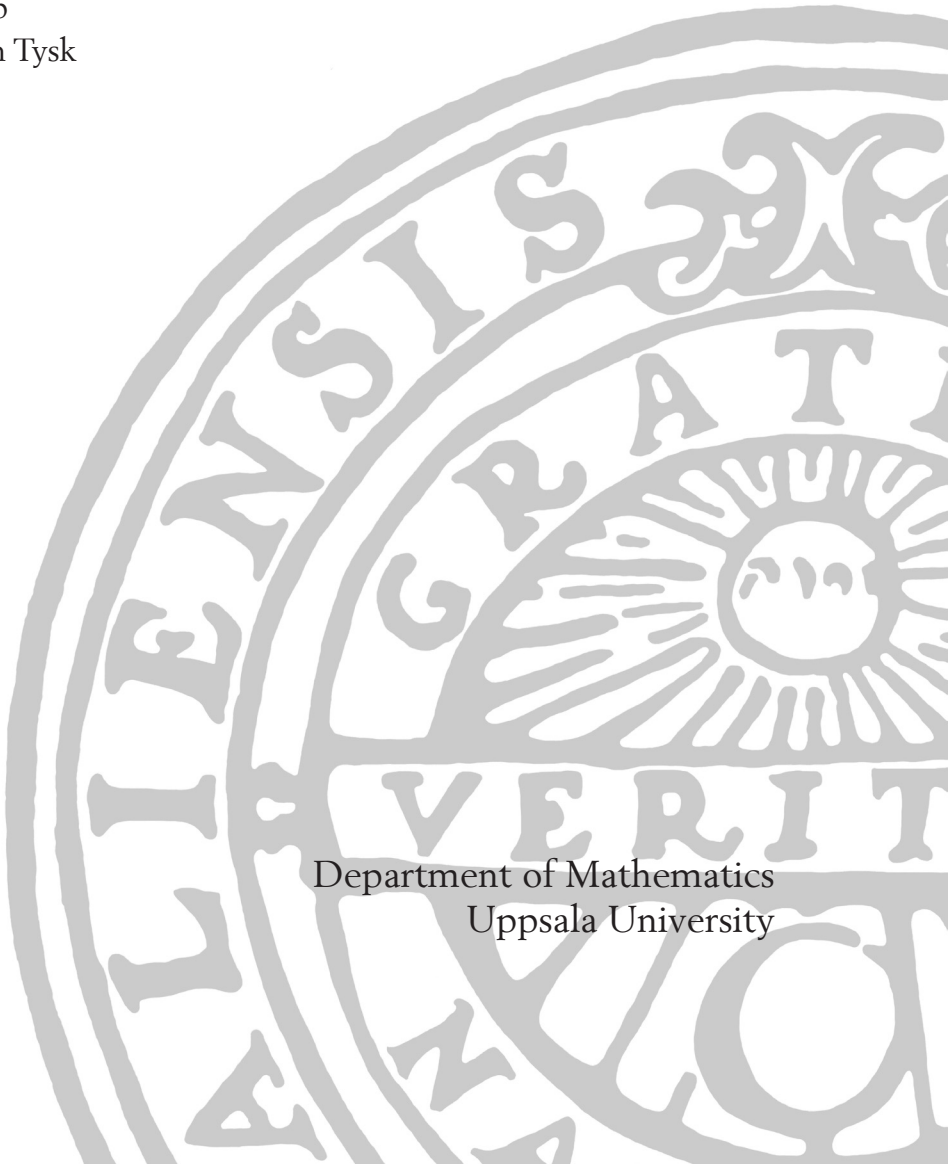
U.U.D.M. Project Report 2011:5

Pricing Barrier Options using Monte Carlo Methods

Bing Wang and Ling Wang

Examensarbete i matematik, 30 hp
Handledare och examinator: Johan Tysk

Maj 2011



Department of Mathematics
Uppsala University

Abstract

Barrier options are cheaper than standard vanilla options, because a zero payoff may occur before expiry. They may match risk hedging needs more closely than ordinary options, which make them particularly attractive to hedgers in the financial market. This paper analyzes the pricing of barrier options using Monte Carlo methods. Four variance reduction techniques are discussed and implemented in the pricing of barrier options. We compare numerical results for option prices from analytical formulas with Monte Carlo simulation where efficiency is improved by different variance reduction methods. From the computational results, we find that the antithetic variates method substantially improves the precision of Monte Carlo estimates. Our results also show that using the ordinary put option as a control variate significantly reduces standard errors. The conditional Monte Carlo and the combined conditional expectation and importance sampling method are found less attractive in our case.

Acknowledgements

We would like to thank our supervisor, Professor Johan Tysk, for his useful suggestions and guidance throughout this study. In this master thesis we study the price estimation for barrier options by Monte Carlo Simulations, using variance reduction methods. Bing Wang writes the theoretical background and mathematical concepts of barrier options, the Monte Carlo simulation and variance reduction techniques. Ling Wang contributes to the paper by introducing and providing the Matlab algorithm and analyzes the computational results. We have shared hours of discussion concerning different problems and enjoyed a great co-operation.

Table of contents

1. Introduction

Barrier options are one of the most widely traded derivatives in the financial markets. They have special characteristics which distinguish them from ordinary options. The payoff of a barrier depends on the path of the underlying asset price. These options are activated or expired when the underlying asset price either hits or does not hit a specified barrier before expiry. One reason that an investor prefers a barrier option to an ordinary vanilla option is that barrier options are generally cheaper than standard options. This is because the asset price has to cross a certain barrier for the option holder to receive the payoff. The other reason is that barrier options may match risk hedging needs more closely than standard options.

Various approaches for pricing barrier options have been developed. One such method is Monte Carlo simulation. Under the Black-Scholes model, barrier options can be considered that the asset price follows the Geometric Brownian Motion. Monte Carlo simulation has been proven to be an effective and simple tool in pricing options. The efficiency and accuracy of estimating option prices can be improved by introducing variance reduction techniques.

In this present thesis, we concentrate on comparing numerical results for option prices from analytical formulas with results of Monte Carlo simulations. To improve the efficiency, a variety of different variance reduction techniques will be considered.

This paper is organized as follows. Section 2 is the literature review on barrier options, the Monte Carlo simulation and variance reduction methods. Section 3 covers the fundamental financial theory. This will include a discussion of the risk-neutral measure, the Black-Scholes model and the option valuation. Section 4 covers the concept and properties of barrier options. The closed form formula for the selected barrier options will be presented. The mathematical concepts and the fundamental theory of Monte Carlo simulation for option payoffs are introduced in Section 5. In Section 6 we price barrier options by Monte Carlo Methods. We introduce various variance reduction methods

which are used to improve the efficiency and accuracy of the simulation. This will include antithetic variates, control variates, conditional Monte Carlo and the combined conditional expectation and importance sampling method. Section 7 compares numerical results obtained from analytical formula with the results of corresponding Monte Carlo simulations. Conclusions are presented in Section 8. Simulation coding is done in Matlab. Codes used for simulation can be found in the Appendix.

2. Literature Review

Black and Scholes (1973) successfully derived the first closed form solution for European options, under the assumption that the stock price follows a lognormal distribution. Merton (1973) provided the first analytical formula for a down and out barrier call option. This was then further extended for all types of standard barriers by Reiner and Rubinstein (1991).

Boyle (1977) first introduced using Monte Carlo simulation to study option pricing, where the payoff was simulated for vanilla options. Hull and White (1987), Johnson and Shanno (1987), Scott (1987), and Figlewski (1992) also used Monte Carlo simulation for analyzing options. This was further extended by introducing variance reduction techniques and Monte Carlo simulation was used for Asian options, Barrier options and American options. Boyle (1977) used antithetic variates and control variates approaches to price European call options. Antithetic variates method was also considered by Fishman and Huang (1983), Rubinstein, Samorodnitsky and Shaked (1985), Hull and White (1987), and Clewlow and Carverhill (1994). Kemna and Vorst (1990) applied the control variates method in pricing Asian options. Broadie and Glasserman (1996) estimated price derivatives in a simulation framework by using control variates. The method was also employed by Clewlow and Carverhill (1994) and Carverhill and Pang (1995) to value options on coupon bonds. Boyle, Broadie and Glasserman (1997) derived the conditional expectation estimator for the price of a barrier option. Ross and Shanthikumar applied the conditional expectation technique to barrier options. They also combined the conditional expectation and importance sampling estimators. The

application of importance sampling in pricing barrier options was described by Boyle. Glasserman, Heidelberger and Shahabuddin (1999) applied the combination of importance sampling and stratified sampling in the Heath-Jarrow-Morton framework. Glasserman and Staum (2001) applied importance sampling in pricing barrier options.

3. Financial Background

3.1 Options

Derivative securities are financial contracts, or financial instruments, whose values are derived from the value of underlying assets. The underlying asset can be a stock, a bond, a foreign currency, an index portfolio, or another derivative security. Derivatives can be used by individuals or financial institutions to hedge risks. There are many types of derivatives such as options, forwards, futures and swaps.

There are two basic kinds of options: call options and put options. A call option is a contract which gives the holder the right--but not the obligation--to buy the underlying asset at a stated price on or before a stated date. On the contrary, a put option is a contract which gives the holder the right--but not the obligation--to sell an asset at a stated price on or before a stated date. The stated price in the contract is called the strike price; the date in the contract is called the maturity or expiry date. The premium is the price paid for an option.

In general, options are defined either as European or American. A European option can only be exercised at the maturity date of the option, otherwise the option simply expires. An American option is more flexible: it can be exercised at any time up to and including the maturity date.

Consider first a European call option with the strike K, the underlying asset's price at maturity S_T and expiry date T, the call option's payoff at maturity is given by:

$$Payoff = \max[0, (S_T - K)] = (S_T - K)^+.$$

Similarly, for a European put option with the strike K, the underlying asset's price at maturity S_T and expiry date T;

$$Payoff == \max[0, (K - S_T)] = (K - S_T)^+.$$

We restrict our studies in this paper to put options, the results for call options are similar.

3.2 Arbitrage and Portfolios

An arbitrage opportunity implies that we make profits without taking any risk. Opportunities for arbitrage are very short-lived as many traders will take advantage of such opportunities and eventually cause the prices to adjust until arbitrage no longer exists. A common principle in the option pricing theory states that there are no arbitrage opportunities or the market is arbitrage-free.

The idea of pricing options with no arbitrage is to construct a portfolio of the stock and a riskless investment to replicate the payoff of the option. The portfolio's value must, by the absence of arbitrage, equal the price of the option at all times. We form the portfolio as follows. It consists of two instruments: m_t shares of stock and b_t units of bond. Therefore the value of the portfolio at time t is

$$V_t = m_t S_t + b_t B_t.$$

With continuous time models, a portfolio consisting of (m_t, b_t) is called self-financing if V_t follows

$$V_t = V_0 + \int_0^t m_u dS_u + \int_0^t b_u dB_u.$$

The change in V_t is only due to changes in the price of assets and the portfolio has no additional cash inflow or outflow during the time period (0, T). Proofs for the following definitions and theorems can be found in Klebaner (2005).

Definition 1 A claim with a payoff X is said to be replicable if there exists a self-financing portfolio $V_t \geq 0$ and $X = V_t$.

The above expressions lead to an important insight in option pricing--the risk-neutral valuation principle.

Theorem 1 Suppose there is a probability measure Q, such that the discounted stock process $Z_t = S_t/B_t$ is a Q-martingale. Then for any replicating trading strategy, the discounted value process V_t/B_t is also a Q-martingale.

This probability measure Q is called risk-neutral probability measure or an equivalent martingale measure. We denote the original probability measure by P.

Definition 2 Two probability measures P and Q are called equivalent if they have same null sets, that is, for any set A with $P(A) = 0$ we also have $Q(A) = 0$ and vice versa.

Theorem 2 (First Fundamental Theorem) A market model does not have arbitrage opportunities if and only if there exists a probability measure Q, equivalent to P, such that the discounted stock process $Z_t = S_t/B_t$ is Q-martingale.

A market model is complete if any claim is replicable and the martingale probability measure Q is unique.

Theorem 3 The price of a claim with payoff X at time $t = 0$ is given by $\frac{1}{B_T} E_Q(X)$.

E_Q denotes that the expected value of the option at date t is taken under the equivalent martingale probabilities. Theorem 3 states that the value of an option at date 0 is the discounted expected value of the payoff. This theorem is central to pricing options.

3.3 Brownian Motion

To model the price process of the underlying asset, we have to choose a stochastic process that satisfies its properties. Financial mathematicians usually consider the price of the underlying asset is described by Brownian Motion.

A stochastic process W is called a Brownian Motion Process if the following properties hold:

- Normal increments: $W_t - W_s$ has normal distribution with mean 0 and variance $t - s$.
- Independent increments: $W_t - W_s$ is independent of the past.
- Continuity of paths: W_t is a continuous function of t.

A Generalized Brownian Motion for a variable S can be defined as $dS_t = adt + bdW_t$, where a and b are constants. In order to describe the price process of a stock, the generalized Brownian motion can be modified to satisfy the basic properties of the stock price, so we have

$$dS_t = \mu S_t dt + \sigma S_t dW_t.$$

The above equation is also known as Geometric Brownian Motion, where μ and σ are constant parameters referred to as drift the volatility

3.4 Black-Scholes Model

Recall that we are working under the assumption of risk neutrality, where no arbitrage opportunities exist. The basic benchmark model for pricing security derivatives is the Black-Scholes model. It was developed by Black and Scholes (1973) and Merton (1973).

The Black-Scholes model consists of two assets with dynamics. First we assume that the stock price is given by

$$dS_t = \mu S_t dt + \sigma S_t dW_t.$$

where μ is the drift, σ is the volatility, W_t is the standard Brownian Motion under the probability measure P. The bond price Bt is

$$dB_t = rB_t dt.$$

where r is the riskless interest rate.

Since the Black-Scholes Model is a complete market model and has no arbitrage opportunities, there exists a unique equivalent martingale measure *Q*. Under the *Q* measure, we have the following stochastic differential equation for S_t instead:

$$dS_t = rS_t dt + \sigma S_t dW_t .$$

where W_t is a standard Brownian motion under the risk-neutral probability measure *Q*. The solution of the above equation is given by:

$$S_t = S_0 \exp((r - \frac{1}{2}\sigma^2)t + \sigma W_t) .$$

3.5 Pricing Vanilla Options

Options with no special features are referred to vanilla options. The Black-Scholes formula for a European call option was derived by Black and Scholes (1973).

$$C_0 = S_0 N(h) - e^{-rt} KN(h - \sigma T),$$

where $h = \left[\ln(S_0 / K) + \left(r + \frac{1}{2}\sigma^2 \right) \right] \Big/ \sigma\sqrt{T}$ and $N(\cdot)$ is the cumulative normal distribution function.

An extension of the Black-Scholes formula is the put-call parity. From this equation we can easily calculate the price of the put option, knowing the price of the call option.

$$S_t + P_t - C_t = Ke^{-r(T-t)},$$

where C_t is the call price and P_t is the put price.

4. Barrier Options

4.1 Exotic and Path-Dependent Options

Exotic options refer to options that depend on the behavior of the price process during their lifetimes. Unlike a vanilla option, the payoff depends not only on the underlying asset price at expiration, but also on its whole path. There are many types of exotic options, such as Asian options, Barrier options and Lookback options. Each has their own unique characteristics. Exotic options in most cases offer cheaper premium prices than

standard vanilla options. Path-dependent options are heavily monitored before expiration. They are of particular interest in learning and testing numerical methods.

4.2 Barrier Options

A barrier option is a type of path-dependent option where the payoff is determined by whether or not the price of the stock crosses a certain level S_b during its life. There are two general types of barrier options, 'in' and 'out' options. In knock-out options, the contract is canceled if the barrier is crossed throughout the whole life. Knock-in options on the other hand are activated only if the barrier is crossed. The relationship between the barrier S_b and the current asset price S_0 indicates whether the option is an up or down option. If $S_b > S_0$, we have an up option; if $S_b < S_0$, we have a down option. Combining these features with the payoffs of call and put options, we can define an array of barrier options.

For example, a down-and-out put option is a put option that becomes worthless if the asset price falls below the barrier S_b. Therefore the risk for the option writer is reduced. It is reasonable to expect that a down-and-out put option is cheaper than a vanilla one, since it may expire worthless if the barrier is hit while the vanilla option would have paid off.

For a given set of parameters, we can combine an in and an out option of the same type to replicate an ordinary vanilla option. This is due to the fact that when one option gets knocked out, the other is knocked in. Therefore holding both a down-and-out and a down-and-in put option is equivalent to holding a vanilla put option. The parity relationship can be described as:

$$P = P_{di} + P_{do},$$

where P is the price of the vanilla put, and P_{di} is the price for the down-and-in option and P_{do} is the price of the down-and-out options.

It is worth pointing out the relationship between the option price and the barrier level. For a knock-out option, increasing the absolute difference between the barrier level and the

initial spot price has a positive effect on the option value, since the probability of knocking out tends to zero as the absolute difference increases. The value of the option converges to the value of an ordinary vanilla option. It is opposite for the knock-in option: increasing the absolute difference reduces the option value since the probability of knocking in approaches zero.

Barrier options are very sensitive to volatility. For knock-out options, increased volatility makes knock-out more probable and therefore leads to a reduction on the option value. The price of knock-in options increases with increased volatility, since knock-in becomes more probable.

The barrier might be monitored continuously or discretely. Analytical pricing formulas are available for certain continuously monitored barrier options. Consider a down-and-out put with strike price K, a barrier S_b, expiring in T. We have the following formula

$$P = Ke^{-rT}\left\{N(d_4) - N(d_2) - a\left[N(d_7) - N(d_5)\right]\right\} - S_0\left\{N(d_3) - N(d_1) - b\left[N(d_8) - N(d_6)\right]\right\},$$

where

$$a = \left(\frac{S_b}{S_0}\right)^{-1+2r/\sigma^2}$$

$$b = \left(\frac{S_b}{S_0}\right)^{1+2r/\sigma^2}$$

and

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

$$d_2 = \frac{\log(S_0/K) + \left(r - \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

$$d_3 = \frac{\log(S_0/S_b) + \left(r - \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

$$d_4 = \frac{\log(S_0/S_b) + \left(r - \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

$$d_5 = \frac{\log(S_0 / S_b) - \left(r - \sigma^2 / 2\right)T}{\sigma\sqrt{T}}$$

$$d_6 = \frac{\log(S_0 / S_b) - \left(r + \sigma^2 / 2\right)T}{\sigma\sqrt{T}}$$

$$d_7 = \frac{\log(S_0 K / S_b^2) - \left(r - \sigma^2 / 2\right)T}{\sigma\sqrt{T}}$$

$$d_8 = \frac{\log(S_0 K / S_b^2) - \left(r + \sigma^2 / 2\right)T}{\sigma\sqrt{T}}.$$

The discrete price converges to the continuous price as the monitoring frequency increases, suggesting that we can adjust the continuous formula to obtain an approximation to the discrete monitoring price. The idea is using the analytical pricing formula of continuous barrier options but shifting the barrier to correct for discrete monitoring. The shift is determined by the monitoring frequency δt, the asset volatility σ, and a constant $\beta \approx 0.5826$.[1]

Theorem 4 Let $V_m(S_b)$ be the price of a discretely monitored knock-in or knock-out down call or up put with barrier S_b. Let $V(S_b)$ be the price of the corresponding continuously monitored barrier option. Then

$$V_m(S_b) = V\left(S_b e^{\pm\beta\sigma\sqrt{\delta t}}\right) + o\left(\frac{1}{\sqrt{m}}\right),$$

where the sign $\pm$ depends on the option type. We take the minus sign for a down put, and the plus sign for an up put. The term β derives from the Riemann zeta function ζ:

$$\beta = -\zeta\left(\frac{1}{2}\right)\Big/\sqrt{2\pi} \approx 1.4604/\sqrt{2\pi} \approx 0.5826.$$

Thus, to use the continuous price as an approximation to the discrete price, we should correct the barrier as follows:

$$S_b \Rightarrow S_b e^{\pm 0.5826\sigma\sqrt{\delta t}},$$

[1] For more detailed information, readers are referred to Mark Broadie, Paul Glasserman. 1997. A Continuity correction for Discrete Barrier Options.

Discrete monitoring lowers the probability of crossing the barrier compared with continuous barrier monitoring. We should expect that the price for a discretely monitored down-and-out option is increased, since hitting the barrier is less likely.

5. Monte Carlo Simulation

5.1 Basic Concepts

Simulation techniques are important tools to deal with path dependent payoffs or distributions with no analytic expression. Monte Carlo simulation may be used for pricing derivative securities, estimating value at risk and simulating hedging strategies. The relative ease of use and flexibility are its main advantages. The disadvantage of Monte Carlo simulation is its computational burden, which may be partially solved by variance reduction methods. We use Monte Carlo simulation for simulating option price values and compare them with results obtained from analytical formulas. In this section, we will go through the main concept of Monte Carlo simulation.

Suppose we want to estimate some θ,

$$\theta = E\big(g(X)\big).$$

where $g(X)$ is an arbitrary function. We can randomly generate n independent sample values $X_1, X_2, ..., X_n$ from the probability density function $f(x)$. The estimator of θ is given by:

$$\widehat{\theta} = \frac{1}{n}\sum_{i=1}^{n} g(X_i).$$

By the law of large numbers, we obtain $\frac{1}{n}\sum_{i=1}^{n} g(X_i) \to E(g(X))$ as $n \to \infty$ or $\widehat{\theta} \to \theta$ as $n \to \infty$. That is, this estimate converges to the actual value as the number of observations increases. The sample variance can be written as:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(g(X_i) - \hat{\theta}\right)^2.$$

The central limit theorem ensures that the distribution of $\frac{\hat{\theta}-\theta}{\sqrt{s/n}}$ tends to the standard normal distribution when *n* is large enough. For large *n* we have

$$P\left(\hat{\theta}-z_{1-\frac{\alpha}{2}}\frac{s}{\sqrt{n}}<\theta<\hat{\theta}+z_{1-\frac{\alpha}{2}}\frac{s}{\sqrt{n}}\right)\approx 1-\alpha.$$

We may try to measure the quality of our estimator by considering the standard error $s/\sqrt{n}$. Clearly, one way to improve the accuracy of the estimate is to increase the number *n*. We would need to run 10,000 simulations trials to receive a reduction of the standard deviation by a factor of 100. Another approach is to reduce the size of *s* directly. This may be accomplished by using variance reduction techniques.

5.2 Monte Carlo Simulation for Option Payoffs

When we use the Monte Carlo simulation to price an option, the approach can be summarized into three steps.

- Simulate *n* sample paths of the underlying asset price over the time interval.
- Calculate the payoff of the option for each path.
- Average the discounted payoffs over sample paths.

The first step in pricing barrier options using Monte Carlo methods is to simulate the sample paths of the underlying stock price. In vanilla options, there is really no need for path generation, only the price of the underlying asset at maturity is of concern. But barrier options are path-dependent options--the payoff is determined by whether or not the price of the asset hits a certain barrier during the life of the option. Due to this path-dependency, simulation of the entire price evolution is necessary. To simulate a sample path, we have to choose a stochastic differential equation describing the dynamics of the price. We consider the price of the underlying asset is described by a Geometric Brownian Motion:

$$dS_t=\mu S_t dt+\sigma S_t dW_t. \quad (1)$$

Using Euler scheme, we have

$$S_{t+\delta t}=(1+\mu\delta t)S_t+\sigma S_t\sqrt{\delta t}\varepsilon.$$

By applying Ito's lemma, we receive the following expression:

$$d\log S_t = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dW_t. \quad (2)$$

S_t is log-normally distributed and thereby we have

$$E[\log(S(t)/S(0))] = vt$$
$$Var[\log(S(t)/S(0))] = \sigma^2 t$$
$$E[S(t)/S(0)] = e^{\mu t}$$
$$Var[S(t)/S(0)] = e^{2\mu t}\left(e^{\sigma^2 t} - 1\right)$$

where $v = \mu - \sigma^2/2$. Integrate equation (2) yielding:

$$S_t = S_0 \exp(vt + \sigma\int_0^t dW(\tau)).$$

Let us discretize the time interval (0, T) with a time step δt. From the equation above and the properties of the standard Wiener process, we obtain

$$S_{t+\delta t} = S_t \exp(v\delta t + \sigma\sqrt{\delta t}\varepsilon). \quad (3)$$

where $\varepsilon \sim N(0,1)$ is a standard normal random variable. Based on equation (3), we can therefore generate sample paths for the asset price.

6. Variance Reduction Techniques for Monte Carlo Simulation

In this section, we price barrier options with the help of variance reduction techniques. We have seen in section 4.2 how the analytical formula for continuous monitoring can be adjusted to reflect discrete monitoring. The analytical result will be used to check the result of Monte Carlo simulation. We can see that variance reduction techniques indeed improve the efficiency of Monte Carlo simulation.

6.1 Antithetic Variates

In Monte Carlo simulations, increasing the sample size is computationally costly. So we may focus on reducing the sample variance. The method of antithetic variates works by replacing independent X's with negatively correlated random variables.

Suppose we want to estimate $E(g(X_1, X_2, ..., X_n))$, where $X_1, X_2, ..., X_n$ are independent random variables. Let Y_1 and Y_2 be random variables with the same distribution as $g(X_1, X_2, ..., X_n)$. Then

$$Var(\frac{Y_1 + Y_2}{2}) = \frac{Var(Y_1) + Var(Y_2) + 2Cov(Y_1, Y_2)}{4}.$$

It is obvious that the variance will be reduced if we have the two samples negatively correlated.

Following the idea outlined above, we generate a random draw from the $N(0, 1)$ distribution. By symmetry $Z \sim -Z$ where $Z \sim N(0, 1)$, we set Y_1 from $X_1, ... X_n \sim N(0, 1)$ and Y_2 from $-X_1, ... -X_n \sim N(0, 1)$. Thus, we generate a pair of random variables (Y_1, Y_2) and the estimator by the antithetic variates method is

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \frac{g(X_i) + g(-X_i)}{2}.$$

where X_i and $-X_i$ are the so called antithetic variates. If g is a monotonically increasing or monotonically decreasing function, then $Cov(g(X), g(-X)) \leq 0$. Therefore, we can achieve a variance reduction by using this method.

6.2 Control Variates

The control variates method is another way to reduce the variance and generate more accurate results. Suppose we want to estimate $\theta = E(X)$ and there is another random variable Y, which is correlated with X in some sense, with a known expected value $E(Y)$. The variable Y is called the control variate. We can define a random variable Z as

$$Z = X + c(Y - E(Y)),$$

where $c \in \Re$. Clearly we have $E(Z) = E(X) + c(E(Y) - E(Y)) = E(X)$, hence we can apply Monte Carlo simulation to Z instead of $E(X)$. We also have

$$Var(Z) = Var(X + cY) = Var(X) + 2cCov(X, Y) + c^2 Var(Y).$$

As we want to minimize the variance we have to choose the optimal value of *c*. Simple calculus gives

$$c_{\min} = \frac{-Cov(X,Y)}{Var(Y)}.$$

In practice, *Cov*(*X*,*Y*) and *Var*(*Y*) have to be estimated using Monte Carlo simulation, since they are usually unknown.

By using the control variates method, we can use Monte Carlo simulation to price the options with no closed-form formulas. If we want to price a down-and-out put option, a suitable control variate could be a plain vanilla put option. Both its expected value and variance can easily be calculated. We can get the put value through the Black-Scholes formula.

To estimate the covariance between the down and out put price and the plain vanilla put price, we have to run a set of pilot replications. The value of *c* that minimizes the variance can be calculated using the formula mentioned above. We can then obtain the barrier price using the previously obtained value of *c*.

6.3 Conditional Monte Carlo

Consider a down-and-in put option whose price is P_{di}. The conditional Monte Carlo is based on the following idea: we shall stop the simulation once the stock price hits the barrier and then use the Black-Scholes formula to compute the price of a put option instead of simulating the price path until expiry. We reduce the variance since we are doing part of the work analytically and leaving less to be done through Monte Carlo simulation. Because of the following parity relationship,

$$P_{do} = P - P_{di}.$$

we can easily get the price of the corresponding knock-out option. Assume the stock prices are observed at discrete time steps:

$$S = \{S_1, S_2, ..., S_M\}.$$

Based on this path, the price of the option is the discounted expected payoff

$$P_{di} = e^{-rT} E\left[I(S)(K - S_M)^+\right],$$

where the indicator function I is

$$I(S) = \begin{cases} 1 & if\ S_j < S_b\ \ for some\ j \\ 0 & otherwise. \end{cases}$$

Now let j^* be the index of the time instant at which the stock crossed the barrier. The option is activated at time $j^*\delta t$, and we can treat it as a vanilla put from now on. Conditional on the crossing time $t^* = j^*\delta t$ and the price S_{j^*} , we may use the Black-Scholes formula to estimate the put option. If the stock price does not hit the barrier before the expiration, assume $j^* = M + 1$. On the other hand, if the barrier is crossed before maturity, we have

$$E\left[I(S)(K - S_M)^+ \middle| j^*, S_{j^*}\right] = e^{r(T-t^*)} B_p(S_{j^*}, K, T - t^*),$$

where $B_p(S_{j^*}, K, T - t^*)$ is the Black-Scholes price of a vanilla put option with initial stock price S_{j^*} , strike price K, and time to maturity $T - t^*$; discounting the price from maturity back to crossing time gives the estimator for the put value. So, the conditional Monte Carlo estimator simulates n stock price paths and averages over the values:

$$I(S)e^{-rt^*} B_p(S_{j^*}, K, T - t^*).$$

If the barrier is hit, we record the crossing time and the stock price observed at the time when the option has been activated. If the barrier is not hit by the expiry, the estimator is simply zero.

6.4 Importance Sampling

The idea of importance sampling is to change the probability distribution of the stock so that the probability of crossing the barrier is increased.

Suppose we want to estimate $\theta = E_f[h(X)] = \int h(x) f(x) dx$, where the expectation using the density $f(x)$ and $h(X)$ is a function of the random variable X. If we know another density $g(x)$ that satisfies the condition $f(x) = 0$ whenever $g(x) = 0$, we may write

$$\theta = \int h(x) \frac{f(x)}{g(x)} g(x) dx = E_g\left[h(X) \frac{f(X)}{g(X)} \right],$$

where E_g indicates the expected value is taken with respected to another density function. $\frac{f(X)}{g(X)}$ is used to compensate for the introduction of the new distribution, and it is typically called the likelihood ratio.

Letting $h^*(X) = h(x) \frac{f(x)}{g(x)}$, we can estimate θ as

$$\begin{aligned} E_f[h(X)] &= \int h(x) f(x) dx = \int \frac{h(x) f(x)}{g(x)} g(x) dx \\ &= \int h^*(x) g(x) dx = E_g[h^*(X)], \end{aligned}$$

This gives us two estimators $h(X)$ with respect to the density $f(x)$ and $h^*(X)$ with respect to the density $g(x)$. The two estimators have the same expectation.

Obtaining a variance reduction by importance sampling depends largely on the choice of the density $g(x)$. The variance of the importance sampling estimator $h^*(X)$ is given by

$$\begin{aligned} Var_g[h^*(X)] &= \int h^*(x)^2 g(x) dx - \theta^2 \\ &= \int h^2(x) \frac{f(x)}{g(x)} f(x) dx - \theta^2, \end{aligned}$$

and the variance of the original estimator $h(X)$ is given by

$$Var_f[h(X)] = \int h^2(x) f(x) dx - \theta^2.$$

The difference between the two variances is

$$\Delta Var = Var_f[h(X)] - Var_g[h^*(X)] = \int h^2(x) \left[1 - \frac{f(x)}{g(x)} \right] f(x) dx.$$

In order to ensure that we have a variance reduction, the integral should be positive. We would like to select a density $g(x)$ such that

$$f(x) > g(x) \text{ when } h^2(x)f(x) \text{ is small,}$$
$$f(x) < g(x) \text{ when } h^2(x)f(x) \text{ is large.}$$

The conditional expectation and importance sampling methods can be combined. We can apply the importance sampling until the stock price crosses the barrier, and the conditional expectation estimator can be used to compute the option price after the barrier is crossed.

In order to generate an asset price path S, we generate a normal variate Z_j with expected value

$$v = (r - \frac{\sigma^2}{2})\delta t$$

and variance $\sigma^2 \delta t$ for each time step. Let Z be the vector of the normal variates, and let $f(Z)$ be its joint density. To make the crossing barrier more likely, we use the modified expected value

$$v - b.$$

Let $g(Z)$ be the tilted density. Combining importance sampling with the conditional expectation, we have:

$$E_g\left[\frac{f(Z)I(S)(K-S_M)^+}{g(Z)}\middle| j^*, S_{j^*}\right] = \frac{f(z_1,\ldots,z_{j^*})}{g(z_1,\ldots,z_{j^*})} E_g\left[\frac{f(Z_{j^*+1},\ldots,Z_M)}{g(Z_{j^*+1},\ldots,Z_M)} I(S)(K-S_M)^+ \middle| j^*, S_{j^*}\right]$$
$$= \frac{f(z_1,\ldots,z_{j^*})}{g(z_1,\ldots,z_{j^*})} E_f\left[I(S)(K-S_M)^+ \middle| j^*, S_{j^*}\right]$$
$$= \frac{f(z_1,\ldots,z_{j^*})}{g(z_1,\ldots,z_{j^*})} e^{r(T-t^*)} B_p(S_{j^*}, K, T-t^*).$$

Then we have to find the likelihood ratio which is used to correct the change in probability measure. The joint density function of a multivariate normal with expected value μ is given by:

$$f(z) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(z-\mu)^T \Sigma^{-1}(z-\mu)}$$

where Σ is the determinant of the covariance matrix. Since the normal variates Z_j are mutually independent, the covariance matrix is diagonal matrix with elements $\sigma^2\delta t$, and the vector of the expected values has components

$$\mu = (r - \frac{\sigma^2}{2})\delta t$$

for the density $f(Z)$ and $\mu - b$ for the density $g(Z)$.

$$\begin{aligned}
\frac{f(z_1,...,z_{j^*})}{g(z_1,...,z_{j^*})} &= \exp\left\{-\frac{1}{2}\sum_{k=1}^{j^*}(\frac{z_k-\mu}{\sigma\sqrt{\delta t}})^2\right\}\exp\left\{\frac{1}{2}\sum_{k=1}^{j^*}\left(\frac{z_k-\mu+b}{\sigma\sqrt{\delta t}}\right)^2\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2\delta t}\sum_{k=1}^{j^*}\left[(z_k-\mu)^2-(z_k-\mu+b)^2\right]\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2\delta t}\sum_{k=1}^{j^*}\left[-2(z_k-\mu)b-b^2\right]\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2\delta t}\left[-2b\sum_{k=1}^{j^*}z_k+2j^*\mu b-j^*b^2\right]\right\} \\
&= \exp\left\{\frac{b}{\sigma^2\delta t}\sum_{k=1}^{j^*}z_k-\frac{j^*b}{\sigma^2}(r-\frac{\sigma^2}{2})+\frac{j^*b^2}{2\sigma^2\delta t}\right\}.
\end{aligned}$$

In practice, we generate the normal variates with expected value ($v - b$), and multiply the conditional estimator by the likelihood ratio[2].

[2] For more detailed information, readers are referred to Paolo Brandimarte. 2006. Numerical Methods in Finance and Economics.

7. Computational Results and Analysis

The results for a down-and-out put option priced using Monte Carlo simulation is presented for discussion, where $S_b < S_0, S_b < K$.

Table 1 Crude Monte Carlo simulation results for a down-and-out put option

Initial value S_0	Barrier value S_b	Accurate value	*n*	Option values by Crude Monte Carlo	Standard error by Crude Monte Carlo
50	40	0.6264	*n*=100 *n*=1000 *n*=10000	0.8060 0.7126 0.7101	0.1862 0.0556 0.0171
55	40	0.4192	*n*=100 *n*=1000 *n*=10000	0.3565 0.4706 0.4607	0.1299 0.0457 0.0140
45	40	0.6054	*n*=100 *n*=1000 *n*=10000	0.9598 0.8082 0.7306	0.2058 0.0596 0.0174
50	45	0.0629	*n*=100 *n*=1000 *n*=10000	0.0293 0.0995 0.0899	0.0296 0.0148 0.0046
50	35	1.4404	*n*=100 *n*=1000 *n*=10000	1.5444 1.5423 1.5063	0.2808 0.0936 0.0288
50	30	1.8136	n=100 n=1000 n=10000	1.6537 1.7509 1.8689	0.3344 0.1024 0.0343

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,10000$

Table 1 gives the results of the crude Monte Carlo simulation for the down-and-out put option with the given parameters. We compare price values with different initial spot prices and barrier levels. Three sample sizes are considered: *n* = 100, 1000, and 10000. The accuracy of the estimate will be represented by the standard error.

We can see that the results of crude Monte Carlo are close to the corresponding accurate option values obtained from the analytical formula. The figures demonstrate that Monte Carlo methods can be used for barrier option pricing.

For the fixed barrier value $S_b = 40$, the down-and-out put option has the lowest price value when the initial spot price has the highest value $S_0 = 55$. The value of the down-and-out put option decreases as S_0 increases. Given the fixed initial spot price $S_0 = 50$, the value of the down-and-out put option increases as we lower the barrier level from 45 to 30. The down-and-out put option increases in price value as S_b decreases. Thus, the relationship between the option price and barrier level is in agreement with what we have discussed in section 3.2. For a knock-out option, increasing the absolute difference between the barrier level and the initial spot price has a positive effect on the option value. The probability of knocking out approaches to zero as the absolute difference increases, and therefore making the value of the option converge to an ordinary vanilla put value.

There are lots of sources of error in a simulation which cannot be eliminated. But one of the simplest ways to improve accuracy is to increase the number of simulations. Table 1 displays that the standard Monte Carlo method benefits greatly when the number of simulations increases. The price value obtained from Monte Carlo simulation approaches to the accurate value as the number of trials increases. The results show that the Monte Carlo simulation converges quickly. In the table above, convergence to within 0.1 of the actual option value is achieved within 1000 simulations. We can conclude that Monte Carlo simulation gives very good estimates with close proximity to the values given by the analytical formula. The price value only changes slightly after n is larger than 1000. As we expected, the standard error decreases as the number of simulations increases.

Table 2 Monte Carlo using variance reduction: Antithetic variates

Initial value S_0	Barrier value S_b	Accurate value	n	Option values by Crude Monte Carlo	Standard error by Crude Monte Carlo	Option values by antithetic variates	Standard error by antithetic variates
50	40	0.6264	n=100	0.8060	0.1862	0.7457	0.1140
			n=1000	0.7126	0.0556	0.6947	0.0383
			n=10000	0.7101	0.0171	0.6927	0.0119
55	40	0.4192	n=100	0.3565	0.1299	0.5956	0.1061
			n=1000	0.4706	0.0457	0.4581	0.0319
			n=10000	0.4607	0.0140	0.4527	0.0099
45	40	0.6054	n=100	0.9598	0.2058	0.7681	0.1360
			n=1000	0.8082	0.0596	0.7228	0.0390
			n=10000	0.7306	0.0174	0.7266	0.0123
50	45	0.0629	n=100	0.0293	0.0296	0.1395	0.0398
			n=1000	0.0995	0.0148	0.1045	0.0108
			n=10000	0.0899	0.0046	0.0858	0.0032
50	35	1.4404	n=100	1.5444	0.2808	1.8714	0.2264
			n=1000	1.5423	0.0936	1.5276	0.0673
			n=10000	1.5063	0.0288	1.4966	0.0203
50	30	1.8136	n=100	1.6537	0.3344	2.0965	0.2550
			n=1000	1.7509	0.1024	1.7871	0.0761
			n=10000	1.8689	0.0343	1.7969	0.0237

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,100000$

Table 2 compares the results by the crude Monte Carlo estimator and the antithetic variates estimator. Note that, the use of antithetic variables has given smaller standard errors for prices, in most case they were reduced significantly. Using the antithetic variates method, the number of simulations required to obtain the same accuracy as the crude Monte Carlo method is reduced. Variance reduction techniques indeed help to cut the need for increasing the number of simulations.

Table 3 Monte Carlo using variance reduction: Control variates

Initial value S_0	Barrier value S_b	Accurate value	n	Option values by Crude Monte Carlo	Standard error by Crude Monte Carlo	Option values with control variate	Standard error with control variate
50	40	0.6264	n=100	0.8060	0.1862	0.7928	0.1761
			n=1000	0.7126	0.0556	0.7419	0.0512
			n=10000	0.7101	0.0171	0.6953	0.0161
55	40	0.4192	n=100	0.3565	0.1299	0.6561	0.1160
			n=1000	0.4706	0.0457	0.4725	0.0370
			n=10000	0.4607	0.0140	0.4572	0.0116
45	40	0.6054	n=100	0.9598	0.2058	0.7637	0.1788
			n=1000	0.8082	0.0596	0.7120	0.0530
			n=10000	0.7306	0.0174	0.6881	0.0169
50	45	0.0629	n=100	0.0293	0.0296	0.0316	0.0239
			n=1000	0.0995	0.0148	0.0865	0.0145
			n=10000	0.0899	0.0046	0.0840	0.0045
50	35	1.4404	n=100	1.5444	0.2808	1.5301	0.1708
			n=1000	1.5423	0.0936	1.4889	0.0633
			n=10000	1.5063	0.0288	1.4832	0.0188
50	30	1.8136	n=100	1.6537	0.3344	1.8927	0.0252
			n=1000	1.7509	0.1024	1.8640	0.0229
			n=10000	1.8689	0.0343	1.8207	0.0096

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,10000, Npilot = 5000$

In Table 3 we compare the control variates estimator with the performance of Monte Carlo simulation. The control variates method results in a price with lower standard error for the same number of paths. The use of control variates quickly results in accurate estimates of the option price. It is clear that the Monte Carlo method using control variates is much more effective at estimating the option price for a given number of simulations.

Table 4 Monte Carlo using variance reduction: Conditional expectations

Initial value S_0	Barrier value S_b	Accurate value	n	Option values by Crude Monte Carlo	Standard error by Crude Monte Carlo	Option values by Conditional Monte Carlo	Standard error by Conditional Monte Carlo
50	40	0.6264	n=100	0.8060	0.1862	0.7391	0.2886
			n=1000	0.7126	0.0556	0.6920	0.0915
			n=10000	0.7101	0.0171	0.6773	0.0291
55	40	0.4192	n=100	0.3565	0.1299	0.1769	0.2387
			n=1000	0.4706	0.0457	0.5295	0.0524
			n=10000	0.4607	0.0140	0.4386	0.0186
45	40	0.6054	n=100	0.9598	0.2058	0.4412	0.3864
			n=1000	0.8082	0.0596	0.7503	0.1199
			n=10000	0.7306	0.0174	0.7156	0.0378
50	45	0.0629	n=100	0.0293	0.0296	0.1850	0.2070
			n=1000	0.0995	0.0148	0.1058	0.0654
			n=10000	0.0899	0.0046	0.0654	0.0207
50	35	1.4404	n=100	1.5444	0.2808	1.6042	0.1941
			n=1000	1.5423	0.0936	1.5919	0.0603
			n=10000	1.5063	0.0288	1.5027	0.0218
50	30	1.8136	n=100	1.6537	0.3344	1.8767	0.0000
			n=1000	1.7509	0.1024	1.8593	0.0174
			n=10000	1.8689	0.0343	1.8391	0.0082

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,10000, bp = 10$

Table 4 shows the simulation results of the conditional Monte Carlo estimators and compares them with the results of the crude Monte Carlo estimator. The conditional expectation Monte Carlo estimator gives more accurate and precise estimates than the crude Monte Carlo estimator. However, the conditional expectation does not yield errors lower than that of the crude Monte Carlo. It appears as if using this variance reduction technique does virtually nothing to reduce the error.

Table 5 Monte Carlo using variance reduction: Combination of conditional expectation and importance sampling

Initial value S_0	Barrier value S_b	Accurate value	n	Option values by Crude Monte Carlo	Standard error by Crude Monte Carlo	Option values with importance sampling	Standard error with importance sampling
50	40	0.6264	n=100	0.8060	0.1862	0.7772	0.2852
			n=1000	0.7126	0.0556	0.5588	0.1133
			n=10000	0.7101	0.0171	0.6895	0.0319
55	40	0.4192	n=100	0.3565	0.1299	0.2544	0.1265
			n=1000	0.4706	0.0457	0.4943	0.0234
			n=10000	0.4607	0.0140	0.4623	0.0079
45	40	0.6054	n=100	0.9598	0.2058	0.3958	0.8077
			n=1000	0.8082	0.0596	0.7189	0.3326
			n=10000	0.7306	0.0174	0.6265	0.1125
50	45	0.0629	n=100	0.0293	0.0296	0.1743	0.2147
			n=1000	0.0995	0.0148	0.1021	0.1915
			n=10000	0.0899	0.0046	0.0557	0.0592
50	35	1.4404	n=100	1.5444	0.2808	1.2882	0.0927
			n=1000	1.5423	0.0936	1.4942	0.0219
			n=10000	1.5063	0.0288	1.4910	0.0070
50	30	1.8136	n=100	1.6537	0.3344	1.8308	0.0073
			n=1000	1.7509	0.1024	1.8255	0.0022
			n=10000	1.8689	0.0343	1.8248	0.0013

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,10000, bp = 10$

Table 5 reports the results for the combined conditional expectation and importance sampling estimator and the crude Monte Carlo estimator. From these numerical results, we can conclude that the combined conditional expectation and importance sampling estimator for pricing barrier options is more effective than the crude Monte Carlo estimator. The improvement may be attributed more to the effectiveness of the importance sampling method than to the conditional expectation method.

Table 6 Monte Carlo simulation results using variance reduction techniques

Accurate value	*n*	Option values by Crude Monte Carlo	Option values by antithetic variate	Option values with control variate	Option values by Conditional Monte Carlo	Option values with importance sampling
0.6264	*n*=100	0.8060	0.7457	0.7928	0.7391	0.7772
		(0.1862)	(0.1140)	(0.1761)	(0.2886)	(0.2852)
	n=1000	0.7126	0.6947	0.7419	0.6920	0.5588
		(0.0556)	(0.0383)	(0.0512)	(0.0915)	(0.1133)
	n=10000	0.7101	0.6927	0.6953	0.6773	0.6895
		(0.0171)	(0.0119)	(0.0161)	(0.0291)	(0.0319)
0.4192	*n*=100	0.3565	0.5956	0.6561	0.1769	0.2544
		(0.1299)	(0.1061)	(0.1160)	(0.2387)	(0.1265)
	n=1000	0.4706	0.4581	0.4725	0.5295	0.4943
		(0.0457)	(0.0319)	(0.0370)	(0.0524)	(0.0234)
	n=10000	0.4607	0.4527	0.4572	0.4386	0.4623
		(0.0140)	(0.0099)	(0.0116)	(0.0186)	(0.0079)
0.6054	*n*=100	0.9598	0.7681	0.7637	0.4412	0.3958
		(0.2058)	(0.1360)	(0.1788)	(0.3864)	(0.8077)
	n=1000	0.8082	0.7228	0.7120	0.7503	0.7189
		(0.0596)	(0.0390)	(0.0530)	(0.1199)	(0.3326)
	n=10000	0.7306	0.7266	0.6881	0.7156	0.6265
		(0.0174)	(0.0123)	(0.0169)	(0.0378)	(0.1125)
0.0629	*n*=100	0.0293	0.1395	0.0316	0.1850	0.1743
		(0.0296)	(0.0398)	(0.0239)	(0.2070)	(0.2147)
	n=1000	0.0995	0.1045	0.0865	0.1058	0.1021
		(0.0148)	(0.0108)	(0.0145)	(0.0654)	(0.1915)
	n=10000	0.0899	0.0858	0.0840	0.0654	0.0557
		(0.0046)	(0.0032)	(0.0045)	(0.0207)	(0.0592)
1.4404	*n*=100	1.5444	1.8714	1.5301	1.6042	1.2882
		(0.2808)	(0.2264)	(0.1708)	(0.1941)	(0.0927)
	n=1000	1.5423	1.5276	1.4889	1.5919	1.4942
		(0.0936)	(0.0673)	(0.0633)	(0.0603)	(0.0219)
	n=10000	1.5063	1.4966	1.4832	1.5027	1.4910
		(0.0288)	(0.0203)	(0.0188)	(0.0218)	(0.0070)
1.8136	*n*=100	1.6537	2.0965	1.8927	1.8767	1.8308
		(0.3344)	(0.2550)	(0.0252)	(0.0000)	(0.0073)
	n=1000	1.7509	1.7871	1.8640	1.8593	1.8255
		(0.1024)	(0.0761)	(0.0229)	(0.0174)	(0.0022)
	n=10000	1.8689	1.7969	1.8207	1.8391	1.8248
		(0.0343)	(0.0237)	(0.0096)	(0.0082)	(0.0013)

$K = 50, r = 0.1, \sigma = 0.2, T = 1, m = 100, n = 100,1000,10000, bp = 10, Npilot = 5000$

Table 6 compares the price values and standard errors of the down-and-out put option using the crude Monte Carlo estimator, the antithetic variates estimator, the control variates estimator, the conditional Monte Carlo estimator and the combined conditional expectation and importance sampling estimator. In any case, we can see the convergence of simulation solution to analytical solution as *n* increases.

In general, the antithetic variates estimator and control variates estimator work better than the other two methods. The numerical results for option prices are more accurate from antithetic variates and control variates methods and the standard errors are smaller.

8. Conclusions

In this paper an overview of how to use Monte Carlo simulation for the price estimation for barrier options is provided. The Monte Carlo simulation, which is rather easy to perform, comes with the cost of computational efficiency. We describe four different variance reduction techniques to eliminate standard errors as much as possible. The variance reduction methods can significantly increase the accuracy of the estimates and reduce the number of simulations needed. We find that the results of simulations for the down-and-out barrier option very well agree with the analytical formula prices.

Furthermore, for a knock-out option, increasing the absolute difference between the barrier level and the initial spot price has a positive effect on option value. The probability of knocking out approaches to zero as the absolute difference increases, and therefore making the value of the option converges to an ordinary vanilla put value.

We can conclude that the Monte Carlo simulation converges quickly. Convergence to within 0.1 of the actual option value is achieved within 1000 simulations. The price value only changes slightly after the number of trials is larger than 1000.

From the numerical results, we find that the antithetic variates method substantially improves the precision of Monte Carlo estimates. We get smaller standard errors when

employ this technique. Our results also show that using the ordinary put option as a control variate significantly reduces standard errors. The conditional Monte Carlo and the combined conditional expectation and importance sampling method are found less attractive in our case. The conditional expectation Monte Carlo estimator gives more precise estimates but yields even larger errors than the crude Monte Carlo estimator. As for the combined conditional expectation and importance sampling method, we think the improvement may be attributed more to the effectiveness of the importance sampling method than to the conditional expectation method.

This paper discusses and compares four variance reduction techniques thoroughly which has not been done in any existing literatures. Our results are consistent with Paolo Brandimarte's conclusions. He presents that the use of antithetic variates and control variates leads to more accurate estimations of the option price. Emmanuel R. Salta's findings are a little different from our conclusions. He proves that the conditional expectation estimator for a down-and-in barrier option can reduce variance significantly. The combined conditional expectation and control variates method does not offer any significant error reduction. The importance sampling approach performs better than the crude Monte Carlo approach for both down-and-in and up-and-out barrier options. Giray Okten, Emmanuel Salta and Ahmet Goncu have drawn different conclusions from us. They claim that the combined importance sampling and conditional expectation technique provides the best error reduction in option pricing. Sibel Kaplan indicates that the variance reduction by conditioning is very helpful to improve the quality of estimates. The importance sampling proves useful by increasing the likeliness of crossing the barrier. The results agree with ours very well.

References

Michael C. Fu, Jian-Qiang Hu. 2005. Sensitivity Analysis for Monte Carlo Simulation of Option Pricing.

Jakub Stoklosa. 2007. Studies of Barrier Options and their Sensitivities.

Augusto, Diana. 2003. Pricing and Hedging Exotic Options with Monte Carlo Simulations.

Paolo Brandimarte. 2006. Numerical Methods in Finance and Economics.

Emmanuel Gobet. 2008. Advanced Monte Carlo Methods for Barrier and Related Exotic Options.

Kyoung-Sook Moon. 2008. Efficient Monte Carlo Algorithm for Pricing Barrier Options

Yoon W. Kwon, Suzanne A. Lewis. 2000. Pricing Barrier Option Using Finite Difference Method and Monte Carlo Simulation.

Giray Okten, Emmanuel Salta, Ahmet Goncu. 2007. On Pricing Discrete Barrier Options Using Conditional Expectation and Importance Sampling Monte Carlo.

Emmanuel R. Salta. 2008. Variance Reduction Techniques in Pricing Financial Derivatives.

Martin Haugh. 2004. Variance Reduction Methods.

Patrik Karlsson. 2009. FX Basket Options--Approximation and Smile Prices.

Yiting Lin. 2008. Variance Reduction Algorithm for Pricing Various Options.

Abukar M Ali. Exotic Options: Pricing Path-Dependent single Barrier Option contracts.

Réda Rebib. 2008. Barrier Options Pricing.

Yuh-Dauh Lyuu. 2004. Financial Engineering And Computation.

Sibel Kaplan. 2008. Monte Carlo Methods for Option Pricing.

Paul Glasserman. 2003. Monte Carlo Methods in Financial Engineering.

Trinity. 2003. Pricing of Discrete Barrier Options.

Fima C Klebaner. 2005. Introduction to Stochastic Calculus with applications.

Mark Broadie, Paul Glasserman. 1997. A Continuity correction for Discrete Barrier Options.

Appendix

```
%AssetPaths.m
function SPaths=AssetPaths(S0,mu,sigma,T,NSteps,NRepl)
SPaths = zeros(NRepl, 1+NSteps);
SPaths(:,1) = S0;
dt = T/NSteps;
nudt = (mu-0.5*sigma^2)*dt;
sidt = sigma*sqrt(dt);
for i=1:NRepl
  for j=1:NSteps
    SPaths(i,j+1)=SPaths(i,j)*exp(nudt + sidt*randn);
  end
end

%AssetPathsAV.m
function [SPathsA, SPathsB] = AssetPathsAV(S0,mu,sigma,T,NSteps,NRepl)
SPathsA = zeros(NRepl, 1+NSteps);
SPathsA(:,1) = S0;
SPathsB = zeros(NRepl, 1+NSteps);
SPathsB(:,1) = S0;
dt = T/NSteps;
nudt = (mu-0.5*sigma^2)*dt;
sidt = sigma*sqrt(dt);
for i=1:NRepl
   for j=1:NSteps
      SPathsA(i,j+1) = SPathsA(i,j)*exp(nudt + sidt*randn);
      SPathsB(i,j+1) = SPathsB(i,j)*exp(nudt + sidt*randn);
   end
end
```

```
% DownOutPut.m
function P = DOPut(S0,K,r,T,sigma,Sb)
a = (Sb/S0)^(-1 + (2*r / sigma^2));
b = (Sb/S0)^(1 + (2*r / sigma^2));
d1 = (log(S0/K) + (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d2 = (log(S0/K) + (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d3 = (log(S0/Sb) + (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d4 = (log(S0/Sb) + (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d5 = (log(S0/Sb) - (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d6 = (log(S0/Sb) - (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d7 = (log(S0*K/Sb^2) - (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d8 = (log(S0*K/Sb^2) - (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
P = K*exp(-r*T)*(normcdf(d4)-normcdf(d2) - ...
    a*(normcdf(d7)-normcdf(d5))) ...
    - S0*(normcdf(d3)-normcdf(d1) - ...
b*(normcdf(d8)-normcdf(d6)));

%DownOutPutMC.m
function [P,CI,NCrossed] = DOPutMC(S0,K,r,T,sigma,Sb,NSteps,NRepl)
[Call,Put] = blsprice(S0,K,r,T,sigma);
Payoff = zeros(NRepl,1);
NCrossed = 0;
for i=1:NRepl
   Path=AssetPaths1(S0,r,sigma,T,NSteps,1);
   crossed = any(Path <= Sb);
   if crossed == 0
      Payoff(i) = max(0, K - Path(NSteps+1));
   else
      Payoff(i) = 0;
      NCrossed = NCrossed + 1;
   end
```

```
end
[P,aux,CI] = normfit( exp(-r*T) * Payoff);

%AVDOPutMC.m
function [P,std,CI] = AVDOPutMC(S0,Sb,K,r,T,sigma,NSteps,NRepl)
[Call,Put] = blsprice(S0,K,r,T,sigma);
PayoffA = zeros(NRepl,1);
PayoffB = zeros(NRepl,1);
for i=1:NRepl
   [PathA, PathB] = AssetPathsAV(S0,r,sigma,T,NSteps,1);
   crossedA = any(PathA <= Sb);
   crossedB = any(PathB <= Sb);
   if crossedA == 0
      PayoffA(i) = max(0,K-PathA(NSteps+1));
   end
   if crossedB == 0
      PayoffB(i) = max(0,K-PathB(NSteps+1));
   end
end
Payoff = (PayoffA + PayoffB)./2;
[P, std, CI] = normfit(exp(-r*T)*Payoff);

%CVDOPutMC.m
function [P,std,CI] = CVDOPutMC(S0,Sb,K,r,T,sigma,NSteps,NRepl,NPilot)
Payoff = zeros(NPilot,1);
VanillaPayoff = zeros(NPilot,1);
[aux, muVanilla] = blsprice(S0,K,r,T,sigma);
for i=1:NPilot
   Path = AssetPaths(S0,r,sigma,T,NSteps,1);
   VanillaPayoff(i) = max(0,K-Path(NSteps+1));
   crossed = any(Path <= Sb);
```

```
   if crossed == 0
   Payoff(i) = max(0,K-Path(NSteps+1));
   end
end
VanillaPayoff = exp(-r*T)*VanillaPayoff;
Payoff = exp(-r*T)*Payoff;

covMat = cov(VanillaPayoff, Payoff);
varVanilla = var(VanillaPayoff);
c = -covMat(1,2)/varVanilla;

newPayoff = zeros(NRepl,1);
newVanillaPayoff = zeros(NRepl,1);
for i=1:NRepl
   Path = AssetPaths(S0,r,sigma,T,NSteps,1);
   newVanillaPayoff(i) = max(0,K-Path(NSteps+1));
   crossed = any(Path <= Sb);
   if crossed == 0
      newPayoff(i) = max(0,K-Path(NSteps+1));
   end
end
newVanillaPayoff = exp(-r*T)*newVanillaPayoff;
newPayoff = exp(-r*T)*newPayoff;
CVpayoff = newPayoff + c*(newVanillaPayoff - muVanilla);
[P, std, CI] = normfit(CVpayoff);

%DOPutMCCond.m
function [Pdo,CI,NCrossed] = ...
  DOPutMCCond(S0,K,r,T,sigma,Sb,NSteps,NRepl)
dt = T/NSteps;
[Call,Put] = blsprice(S0,K,r,T,sigma);
```

```
NCrossed = 0;
Payoff = zeros(NRepl,1);
Times = zeros(NRepl,1);
StockVals = zeros(NRepl,1);
for i=1:NRepl
  Path=AssetPaths1(S0,r,sigma,T,NSteps,1);
  tcrossed = min(find( Path <= Sb ));
  if not(isempty(tcrossed))
    NCrossed = NCrossed + 1;
                Times(NCrossed) = (tcrossed-1) * dt;
                StockVals(NCrossed) = Path(tcrossed);
  end
end
if (NCrossed > 0)
  [Caux, Paux] = blsprice(StockVals(1:NCrossed),K,r,...
    T-Times(1:NCrossed),sigma);
  Payoff(1:NCrossed) = exp(-r*Times(1:NCrossed)) .* Paux;
end
[Pdo, aux, CI] = normfit(Put - Payoff);

%DOPutMCCondIS.m
function [Pdo,CI,NCrossed] = ...
  DOPutMCCondIS(S0,K,r,T,sigma,Sb,NSteps,NRepl,bp)
dt = T/NSteps;
nudt = (r-0.5*sigma^2)*dt;
b = bp*nudt;
sidt = sigma*sqrt(dt);
[Call,Put] = blsprice(S0,K,r,T,sigma);

NCrossed = 0;
Payoff = zeros(NRepl,1);
```

```
Times = zeros(NRepl,1);
StockVals = zeros(NRepl,1);
ISRatio = zeros(NRepl,1);
for i=1:NRepl
        vetZ = nudt - b + sidt*randn(1,NSteps);
        LogPath = cumsum([log(S0), vetZ]);
        Path = exp(LogPath);
  jcrossed = min(find( Path <= Sb ));
  if not(isempty(jcrossed))
    NCrossed = NCrossed + 1;
    TBreach = jcrossed - 1;
                Times(NCrossed) = TBreach * dt;
                StockVals(NCrossed) = Path(jcrossed);
    ISRatio(NCrossed) = exp( TBreach*b^2/2/sigma^2/dt +...
      b/sigma^2/dt*sum(vetZ(1:TBreach)) - ...
      TBreach*b/sigma^2*(r - sigma^2/2));
  end
end
if (NCrossed > 0)
  [Caux, Paux] = blsprice(StockVals(1:NCrossed),K,r,...
    T-Times(1:NCrossed),sigma);
  Payoff(1:NCrossed) = exp(-r*Times(1:NCrossed)) .* Paux ...
    .* ISRatio(1:NCrossed);
end
[Pdo, aux, CI] = normfit(Put - Payoff);
```