Efficient Pricing of Barrier Options on High Volatility Assets using Subset Simulation

Keegan Mendonca,∗ Vasileios E. Kontosakos† Athanasios A. Pantelous‡ Konstantin M. Zuev§

January 18, 2018

Abstract

Barrier options are one of the most widely traded exotic options on stock exchanges. They tend to be cheaper than the corresponding vanilla options and better represent investor’s beliefs, but have more complicated payoffs. This makes pricing barrier options an important yet non-trivial computational problem. In this paper, we develop a new stochastic simulation method for pricing barrier options and estimating the corresponding execution probabilities. We show that the proposed method always outperforms the standard Monte Carlo approach and becomes substantially more efficient when the underlying asset has high volatility, while it performs better than multilevel Monte Carlo for special cases of barrier options and underlying assets. These theoretical findings are confirmed by numerous simulation results.

JEL classification: G13, C15

Keywords: Option Pricing; Barrier Options; Path–Dependent Derivatives; Monte Carlo; Subset Simulation; Discretely Monitored

∗Department of Computing and Mathematical Sciences, California Institute of Technology, E. California Blvd., Pasadena, CA 91125 USA. Email: mendoncakeegan@gmail.com
†Department of Econometrics and Business Statistics, Monash University, Wellington Rd, Clayton, Victoria 3800, Australia. Email: vas.kontosak@gmail.com
‡Department of Econometrics and Business Statistics, Monash University, Wellington Rd, Clayton, Victoria 3800, Australia. Email: Athanasios.Pantelous@monash.edu.
§Corresponding author. Department of Computing and Mathematical Sciences, California Institute of Technology, E. California Blvd., Pasadena, CA 91125 USA. Email: kostia@caltech.edu
1 Introduction

A barrier option is among the most actively-traded path-dependent financial derivatives whose payoff depends on whether the underlying asset has reached or exceeded a predetermined price during the option’s contract term (Hull 2009). In the corresponding literature, it is considered a type of exotic option as long as it is more complex than basic American or European options. A barrier option is typically classified as either knock-in or -out. It can be a knock-out, meaning it can expire worthless if the underlying exceeds a certain price. In practice, it limits profits and losses for the holder as well as for the writer, accordingly. Furthermore, it can also be a knock-in, meaning it has no value until the underlying reaches a certain price.

In market practice, more complicated structures exist including options with two barriers, whose asset price trajectory has to remain within a “price band”, with non-fixed barriers, the basket barrier, and the barrier rainbow option, contracts for which the payoff depends on one underlying asset and the barrier is connected to a second underlying asset.

These options tend to be cheaper than the standard ones because they expire more easily and are less likely to be executed (Jewitt 2015). It was estimated that barrier options accounted for approximately half the volume of all traded exotic options (Luenberger and Luenberger 1999). Despite the 2007–08 credit crunch and the subsequent drop in the demand for path-dependent financial instruments, barrier options can still be a useful investment or hedging vehicle when the structure and the risks of the product are comprehensible.

Barrier options can be traded for a number of reasons, using mostly commodities and interest rates as the underlying asset(s). First, barrier options more accurately represent investor’s beliefs than the corresponding plain-vanilla option. Second,
barrier options offer a more attractive risk–reward relation than plain–vanilla options, and their advantage stems from their lower price that reflects the additional risk that the spot price might never reach (knock–in) or cross (knock–out) the barrier throughout its life. Barrier options on high volatility underlying assets can be used in a similar way as cheap deep out–of–the–money options, serving as a hedge to provide insurance in a financial turmoil, given their volatility–dependence (Carr and Chou 2002). Hence, the development of a framework able to deal efficiently with barrier options on high volatility underlying assets tackles an actual problem in computational finance, which to our knowledge has not been explicitly studied in past. According to Andersen et al. (2001), the mean annualized volatility of the thirty stocks in the Dow Jones Industrial Average (DJIA) is approximately equal to 28% (ranging between 22% and 42%) while it is not uncommon to record stocks with volatility levels between 33 and 40%.3

The pricing of barrier options is a challenging problem due to the need to monitor the price of the underlying asset and compare it against the barriers. On one hand, analytical approaches, such as most of the Black-Scholes (BS)–based models, assume continuity in the monitoring of barriers while in practice this happens discretely (Kou 2007). Traders usually set points in time (e.g., daily or weekly) at which they monitor the price of the underlying security to figure out whether the contract is still active or not. This in turn, gives rise to a series of problems, among which probably the most important is that the barrier might be crossed at some point between two consecutive monitor points. On the other hand, pricing discretely monitored barrier options is not a trivial task as in essence we have to solve a multi–dimensional integral of normal distribution functionals, where the dimension of the integral is defined by the number of discrete monitoring points (Fusai and Recchioni 2007).

Computationally, certain barrier options, such as down-and-out options, can be

3There are also other major advantages of barrier options. Namely, there are limits to how much the party selling the option could pay, when in vanilla options, there is no set limit on the potential downside. Furthermore, options are often used for pricing various assets. For example, corporate debt and equity can be modeled using options (Merton 1974). Barrier options, however, better model corporate securities, as a vanilla option only models bankruptcy occurring at the end of the contract, while in a barrier option, bankruptcy can occur at any time (Brockman and Turtle 2003).
priced via the standard *Black–Scholes–Merton* (BSM) model as described in the seminal Merton (1973)’s paper. This idea can be further extended to more complicated barrier options which can be priced using replicating portfolios of vanilla options in a BSM framework (Carr and Chou, 2002). All these approaches, however, suffer from the BSM model’s dependence on a number of assumptions, including amongst others frictionless and efficient markets, perfectly divisible assets and known and constant interest rate and volatility conditions which are not met in real–world trading (Hull, 2009). As a result, the estimates we obtain for the fair value of a contract under the equivalent martingale measure (EMM) are often inaccurate.

While there are other models for barrier options with analytical solutions, such as jump-diffusion models (Kou, 2002; Kou and Wang, 2004), and the constant elasticity of variance (CEV) model (Boyle and Tian, 1999; Davydov and Linetsky, 2001), as well as some other approaches such as the exact analytical (Fusai et al., 2006), the Hilbert transform-based (Feng and Linetsky, 2008) and Fourier-cosine-based (Lian et al., 2017) semi-analytical methods, they depend on a very specific set of assumptions. Another set of methods for pricing barrier options was proposed in Boyle and Tian (1998); Zvan et al. (2000); Zhu and De Hoog (2010); Golbabai et al. (2014). These methods are based on solving partial differential equations (PDEs) that model the claims underlying the barrier option. Although these methods are generally powerful, they depend on being able to accurately model the option with PDEs and cannot be used in all circumstances. As a result, *Monte Carlo simulation* (MCS) is often used for option pricing (Schoutens and Symens, 2003).

The main advantage of MCS over other pricing methods is its model–free property and its non–dependence on the dimension $N$ of the approximated equation. The latter is an important property since as $N \to \infty$ ($\Delta t \to 0$), the price of a

---

4The interested reader is referred to Dupire (1997) and the section ”Pricing and Hedging with Smiles” by B.Dupire, where a model with time–dependent (local) volatility is presented.

5Other approaches used in the pricing of exotic derivatives include the method of lines (Chiarella et al., 2012), where the Greeks are also estimated, robust optimization techniques (Bandi and Bertsimas, 2014), applicable also to American options, finite–difference based approaches (Wade et al., 2007), where a Crank–Nicolson smoothing strategy to treat discontinuities in barrier options is presented, and regime–switching models (Elliott et al., 2014; Rambeerich and Pantelous, 2016).
discretely monitored barrier option converges to that of a continuously monitored one (Broadie et al., 1997). On the other hand, MCS has a serious drawback: it is inefficient in estimating prices of barrier options on high volatility assets. Indeed, high volatility makes it difficult for the asset to remain within barriers, which, in turn, makes a positive payoff — no matter how large it could be — a rare event. As a result any standard MC method will be inaccurate and highly unstable. This motivates the development of more advanced stochastic simulation methods which inherit the universality of MCS, and yet are more efficient in estimating barrier option prices. To overcome this deficiency, other stochastic simulation techniques for speeding up the convergence have been recently proposed, for instance sequential MC (Shevchenko and Del Moral, 2017).

In this study, we develop a novel stochastic simulation method for validating barrier options which is based on the Subset Simulation (SubSim) method, an algorithm originally introduced in Au and Beck (2001) to deal with reliability analysis problems where one needs to estimate small failure probabilities of complex engineered systems. Here, we extend this idea to compute both the execution probabilities and prices of barrier options. We are particularly interested in calculating the fair price for double barrier options on high volatility assets and barriers set near the starting price of the underlying asset. In our framework, the “failure” probability corresponds to the probability of the barrier option to stay alive until maturity (i.e., to be executed at maturity). This setting in a simple MCS results — with an extremely large probability — in asset price trajectories which cross the barriers, rendering the barrier option invalid before maturity.

We show by measuring the coefficient of variation (CV) and the mean squared error (MSE) that the proposed SubSim–based algorithm is an efficient technique for the pricing of such derivatives. In particular, the SubSim estimator has a CV which is $O(|\log p_E|^{d/2})$, where $p_E$ is the execution probability and $d \leq 3$ is a constant. Comparing this against the MC estimator whose CV is $O(p_E^{-1/2})$ and assuming very small values of $p_E$, we can easily see that the latter increases at a dramatically faster pace compared to the SubSim estimator. Moreover, the MSE of the created SubSim estimator is $O(|\log p_E|^{-k/2})$, which decreases for increasing $p_E$. Finally, we
compare our results against the Multi–level Monte Carlo (MLMC) \cite{giles2008multilevel} approach and show that for very small values of the option’s survival probability \( p_E \) the SubSim estimator outperforms the MLMC estimator in terms of the observed CV. Thus our method can be seen as an alternative to price path–dependent options which also completes MLMC for special cases of underlying assets.

This paper begins with the introduction of the SubSim method and an outline of its structure is provided. The main theorem and its proof are then presented. This establishes the limiting behaviour of the MSE and the computational complexity for a broad category of applications. Then, some refinements to the method and its implementation, and the effects of different payoff functions and numerical discretisations are discussed. Finally, numerical results and comparisons with the standard MC and the MLMC methods are presented to provide support for the theoretical analysis, and directions for further research are outlined.

2 Subset Simulation Method

Subset Simulation is a general method for computing small probabilities of rare events. It was originally introduced by \cite{au2001subset} for estimating reliability of structural systems subject to natural disastrous events, and then extended by \cite{zuev2015subset} to complex networks.\footnote{For more details, the reader is referred to \cite{au2014subset}.} Suppose that the time–evolution of the dynamic system under study (e.g. evolution of the asset price \( S_n \)) is modeled by the following discrete model:

\[
X_n = F(X_{n-1}, U_n), \quad n = 1, \ldots, N, \tag{1}
\]

where \( X_n \) is the state of the system at time \( t_n \), \( X = (X_1, \ldots, X_N) \) is the trajectory of the system state, \( U_n \) is a random input at time \( t_n \), and \( F \) is a certain deterministic function that governs the evolution. Let \( g(X) \) be a performance function that represents some quantity of interest (e.g. the maximum value of the asset price \( g(S) = \max_{n=1,\ldots,N} S_n \)). We say that a target event \( E \) occurs if \( g(X) \) exceeds a critical
threshold $\alpha$:

$$E = \{ U = (U_1, \ldots, U_N) : g(X(U)) \geq \alpha \} \subset \mathbb{R}^N. \quad (2)$$

The central idea behind SubSim is to break down the rare event of interest $E$ into a series of “less rare” events that have easier-to-compute probabilities. This idea is implemented by considering a collection of nested subsets starting from the entire input space $\mathbb{R}^N$ and finishing at the target rare event,

$$\mathbb{R}^N = E_0 \supset E_1 \supset \ldots \supset E_L \equiv E. \quad (3)$$

The intermediate events $E_i$ can be defined by simply repeatedly relaxing the value of the critical threshold $\alpha$ in (2),

$$E_i = \{ U = (U_1, \ldots, U_N) : g(X(U)) \geq \alpha_i \}, \quad \alpha_1 < \alpha_2 < \ldots < \alpha_L \equiv \alpha. \quad (4)$$

Given the sequence (3), the small probability $p_E$ of rare event $E$ can be written as a product of conditional probabilities:

$$p_E = \mathbb{P}(E_L) = \mathbb{P}(E_L|E_{L-1})\mathbb{P}(E_{L-1})$$

$$= \mathbb{P}(E_L|E_{L-1})\mathbb{P}(E_{L-1}|E_{L-2})\mathbb{P}(E_{L-2}) = \ldots = \prod_{i=1}^{L} \mathbb{P}(E_i|E_{i-1}). \quad (5)$$

By choosing the intermediate thresholds $\alpha_i$ appropriately (in the actual implementation of SubSim described below, $\alpha_i$ are chosen adaptively on the fly), we can make all conditional probabilities $\mathbb{P}(E_i|E_{i-1})$ large enough, and estimate them efficiently by MC-like simulation methods. In fact, the first factor in the right-hand side of (5), $\mathbb{P}(E_1|E_0) = \mathbb{P}(E_1)$, can be directly estimated by MCS:

$$\mathbb{P}(E_1) \approx \frac{1}{m} \sum_{i=1}^{m} I_{E_1} \left(U^{(i)} \right), \quad U^{(1)}, \ldots, U^{(m)} \sim f_U. \quad (6)$$

Estimating the remaining factors $\mathbb{P}(E_i|E_{i-1})$ for $i \geq 2$ is more difficult since this requires sampling from the conditional distribution $f_U(u|E_{i-1}) \propto f_U(u)I_{E_{i-1}}(u)$, which is a nontrivial task, especially at later levels, where $E_{i-1}$ becomes a rare event. In SubSim, this is achieved by using the so-called modified Metropolis algorithm.
(MMA) (Au and Beck, 2001; Zuev and Katafygiotis, 2011), which belongs to a large family of Markov chain Monte Carlo algorithms (Liu, 2001; Robert and Casella, 2004) for sampling from complex probability distributions. The MMA algorithm is a component-wise modification of the original Metropolis algorithm (Metropolis et al., 1953), which is specifically tailored for sampling in high dimensions, where the original algorithm is known to perform poorly (Katafygiotis and Zuev, 2008).

To sample from \( f_{U|E_{i-1}}(u) \), MMA generates a Markov chain whose stationary distribution is \( f_{U|E_{i-1}}(u) \). The key difference between MMA and the original Metropolis algorithm is how the “candidate” state of a Markov chain is generated. Namely, if we let \( U^{(j)} \in E_{i-1} \) be the current state, then the next state \( U^{(j+1)} \) is generated as follows:

1. Generate a candidate state \( \Upsilon = (\Upsilon_1, \ldots, \Upsilon_N) \):
   
   (a) For each \( k = 1, \ldots, N \), generate \( \Psi_k \sim q(\psi|U_k^{(j)}) \), where \( q \) is a symmetric, \( q(\psi|u) = q(u|\psi) \), univariate proposal distribution, e.g., Gaussian distribution centered at \( U_k^{(j)} \), the \( k \)th component of \( U^{(j)} \).
   
   (b) Compute the acceptance probability:
   
   \[
   a_k = \min \left\{ 1, \frac{f_k(\Psi_k)}{f_k(U_k^{(j)})} \right\},
   \]

   where \( f_k \) is the marginal PDF of \( U_k \), \( f_U(u) = \prod_{k=1}^{N} f_k(u_k) \), and \( U_1, \ldots, U_N \) are assumed to be independent.

   (c) Set
   
   \[
   \Upsilon_k = \begin{cases} 
   \Psi_k, & \text{with probability } a_k, \\
   U_k^{(j)}, & \text{with probability } 1 - a_k.
   \end{cases}
   \]

2. Accept or reject the candidate state:

   \[
   U^{(j+1)} = \begin{cases} 
   \Upsilon, & \text{if } \Upsilon \in E_{i-1}, \\
   U^{(j)}, & \text{if } \Upsilon \notin E_{i-1}.
   \end{cases}
   \]

Using the detailed balance equation, it can be shown that if \( U^{(j)} \) is distributed according to the target distribution, \( U^{(j)} \sim f_{U|E_{i-1}}(u) \), then so is \( U^{(j+1)} \), and
\( f_U(u|E_{i-1}) \) is thus indeed the stationary distribution of the Markov chain generated by MMA \( \text{(Au and Beck 2001).} \)

Now, having introduced all key elements, we are ready to describe how SubSim estimates the small probability \( p_E \) of rare event \( E \). The method starts by generating \( m \) MC samples \( U^{(1)}, \ldots, U^{(m)} \sim f_U \) and computing the corresponding system trajectories \( X^{(1)}, \ldots, X^{(m)} \) via (1) and performance values \( g_U^{(i)} = g(X^{(i)}) \). Without loss of generality, we can assume that

\[
g_U^{(1)} \geq g_U^{(2)} \geq \ldots \geq g_U^{(m)}.
\]  

(10)

Indeed, to achieve this ordering, we can simply renumber the samples accordingly. Since \( E \) is a rare event, all \( U^{(i)} \notin E \) with large probability. The ordering \( (10) \) means however that, in the metric induced by the performance function, \( U^{(1)} \) is the closest sample to \( E \), \( U^{(2)} \) is the second closest, etc. Let’s define the first intermediate threshold \( \alpha_1 \) as the average between the performance values of the \( \tilde{m}^{\text{th}} \) and \( (\tilde{m}+1)^{\text{th}} \) system trajectories, where \( \tilde{m} = \beta m \) with \( \beta \in (0, 1) \):

\[
\alpha_1 = \frac{g_U^{(\beta m)} + g_U^{(\beta m+1)}}{2}, \quad 0 < \beta < 1.
\]

(11)

Setting \( \alpha_1 \) to this value has two important corollaries: (i) the MC estimate of \( P(E_1) \) given by (6) is exactly \( \beta \), and (ii) samples \( U^{(1)}, \ldots, U^{(\beta m)} \) are i.i.d. random vectors distributed according to the conditional distribution \( f_U(u|E_1) \).

In the next step, SubSim generates \( \tilde{m} = \beta m \) Markov chains by MMA starting from \( \tilde{m} \) most closest to \( E \) samples \( U^{(1)}, \ldots, U^{(\beta m)} \) as “seeds”:

\[
U^{(i)} = V^{(i,1)} \xrightarrow{\text{MMA}} V^{(i,2)} \xrightarrow{\text{MMA}} \ldots \xrightarrow{\text{MMA}} V^{(i,l)}.
\]

(12)

Since, by construction, all seeds are in the stationary state, \( U^{(i)} \sim f_U(u|E_1), i = 1, \ldots, \tilde{m}, \) so are all Markov chains states \( V^{(i,j)} \sim f_U(u|E_1), j = 1, \ldots, l \). The length of each chain is \( l = 1/\beta \), which makes the total number of states \( \tilde{m}l = m \). To simplify the notation, let’s denote samples \( V^{(i,j)} \) by simply \( V^{(1)}, \ldots, V^{(m)} \).
Next, the second intermediate threshold $\alpha_2$ is similarly defined as follows:

$$\alpha_2 = \frac{g_{\beta m}^{(\beta m)} + g_{\beta m + 1}^{(\beta m + 1)}}{2},$$

(13)

where $g_{\beta}^{(1)} \geq g_{\beta}^{(2)} \geq \ldots \geq g_{\beta}^{(m)}$ are the ordered performance values corresponding to samples $V^{(1)}, \ldots, V^{(m)}$. Again, by construction, $\mathbb{P}(E_2|E_1) \approx \beta$ and $V^{(1)}, \ldots, V^{(\beta m)} \sim f_U(u|E_2)$. The SubSim method, schematically illustrated in Figure [1] proceeds in this way by directing Markov chains towards the rare event $E$ until it is reached and sufficiently sampled. Specifically, it stops when the number $m_E$ of samples in $E$, which a priori $0 \leq m_E \leq m$, is $m_E \geq \beta m$. All but the last factor in the right-hand side of (5) are then approximated by $\beta$ and $\mathbb{P}(E|E_{L-1}) \approx m_E/m$. This results into the following estimate:

$$p_E \approx \hat{p}_{E_{\text{SubSim}}} = \beta^{L-1} \frac{m_E}{m},$$

(14)

where $L$ is the number of subsets in (5) required to reach $E$. The total number of samples used by SubSim is then

$$M = \frac{m}{\text{MCS}} + \frac{m(1-\beta)(L-1)}{\text{MMA}}.$$  

(15)

The CV of the SubSim estimator, derived in [Au and Beck, 2001], scales like a power of the logarithm of $p_E$,

$$\delta(\hat{p}_{E_{\text{SubSim}}}) = \sqrt{\frac{1 + \gamma}{M\beta(|\ln \beta|^d |\ln p_E|^d |\ln p_E|^d)^\gamma}} \propto |\ln p_E|^{d/2},$$

(16)

where $\gamma$ is a constant that depends on the correlation of the Markov chain states and $2 \leq d \leq 3$. Comparing (16) against the CV of a standard MC method [Liu, 2001; Robert and Casella, 2004]

$$\delta(\hat{p}_{E_{\text{MC}}}) = \sqrt{\text{Var}[\hat{p}_{E_{\text{MC}}}] / \mathbb{E}[\hat{p}_{E_{\text{MC}}}]} = \sqrt{\frac{1-p_E}{M p_E}} \propto p_E^{-1/2},$$

(17)

reveals a serious drawback of MC: it is inefficient in estimating small probabilities
of rare events. Indeed, as \( p_E \to 0 \), then
\[
\delta \left( \hat{p}^{MC}_E \right) \approx \frac{1}{\sqrt{M p_E}}. 
\]
This means that the number of samples \( M \) needed to achieve an acceptable level of accuracy is inversely proportional to \( p_E \), and therefore very large, \( M \propto 1/p_E \gg 1 \). Therefore, for rare events, where probabilities are small \( p_E \ll 1 \), the CV of SubSim is significantly lower than that of MC, \( \delta \left( \hat{p}^{SubSim}_E \right) \ll \delta \left( \hat{p}^{MC}_E \right) \). This property guarantees that SubSim produces more accurate (on average) estimates of small probabilities of rare events.

### 3 Complexity Theorem

The statistical properties of Subset Simulation can be formulated in terms of the limiting behavior of the mean-squared error (MSE) and the computational cost (or complexity) of the SubSim estimator \( \hat{P} \). To this end, we formulate the following theorem:

**Theorem 1.** The SubSim estimator \( \hat{P} \) for a functional of the solution \( \hat{S} \) to a given stochastic differential equation (SDE) has

(i) a MSE bounded from above by \( c_1 \delta^2 |\log p_E|^{-k} \),

(ii) with computational cost which has an upper bound of \( c_2 \delta^{-2} |\log p_E|^r \),

where \( c_1, c_2 \) are constants, \( \delta \) is the CV of \( \hat{P} \), \( p_E \) is the probability of positive payoff at maturity and \( r \) a parameter dependent on the correlation between the intermediate execution probabilities.

**Proof.** Using result [16] we have that the squared CV of \( \hat{P} \) is equal to

\[
\delta^2 = \frac{(1 + \gamma)(1 - \beta)}{\beta |\log \beta|^r L m} |\log p_E|^r, \tag{18}
\]

where \( \gamma \) is a constant related to the correlation between the states of the Markov chains used for the sampling at different levels, \( \beta \) is the level probability, \( L \) is the total number of subsets and \( m \) represents the number of samples per subset (the product \( L m \) approximates the total number of samples \( M \) in [16]). The complexity of Subset Simulation, given by the product of the samples per level times the number

11
of simulation levels used is equal to

\[ C = Lm = \frac{(1 + \gamma)(1 - \beta)}{\beta |\log \beta| r \delta^2} |\log p_E|^r = \frac{(1 + \gamma)(1 - \beta)}{\beta \delta^2} |L|^r, \quad (19) \]

by noting that the number of simulation levels \( L \) is chosen as \( L = \log p_E/\log \beta \). Fixing \( \beta \) and treating \( \gamma \) as a known constant we have that

\[ C \propto |L|^r \delta^{-2} \leq c_2 |L|^r \delta^{-2} \quad \text{or} \quad C \leq c_2 \delta^{-2} |\log p_E|^r, \quad (20) \]

which yields the upper bound of the computational complexity, given that \( L \) is \( O(|\log p_E|^r) \) for fixed \( \beta \). Moreover, considering the definition for the coefficient of variation for \( \hat{\delta} \) we have

\[ \delta_{\hat{\delta}} = \sqrt{\text{VAR}[\hat{P}]} = \sqrt{\frac{\text{MSE}[\hat{P}] + \text{BIAS}[\hat{P}, P]^2}{\mathbb{E}[P]}}, \quad (21) \]

Squaring both sides of (21) gives

\[ \delta^2_{\hat{\delta}} = \frac{\text{MSE}[\hat{P}] - \text{BIAS}[\hat{P}, P]^2}{\mathbb{E}[P]^2}, \quad (22) \]

which equivalently can be written as

\[ \text{MSE}[\hat{P}] = \delta^2_{\hat{\delta}} \mathbb{E}[\hat{P}]^2 + \text{BIAS}[\hat{P}, P]^2. \quad (23) \]

Now, we use Propositions 1 and 2 [Au and Beck, 2001] which prove that both the bias and the squared CV \( \delta^2 \) of the estimator \( \hat{P} \) are bounded above by \( c_3/m \). As a result, the first term of the \( \text{MSE} \) is \( O(1/m) \) while the second term is \( O(1/m^2) \) which gives an \( \text{MSE} \) bounded above by \( 1/m \) as for large values of \( m \) it dominates the \( O(1/m^2) \) term.

By equation (18) we also notice that \( \delta^2 \) is \( O(|\log p_E|^r L^{-1} \delta^{-2}) \) from which we obtain \( m = O(|\log p_E|^r L^{-1} \delta^{-2}) \). Setting \( L = \log p_E/\log \beta = O(|\log p_E|) \) and fixing \( \delta^2 \), the number of samples \( m \) becomes \( O(|\log p_E|^k) \) where \( k = r - 1 \leq 3 \) is a new
constant. Consequently, we end up with an MSE bounded from above by

\[ \text{MSE} \triangleq \mathbb{E}[(P - \hat{P})^2] \leq c_1 \frac{1}{|\log p_E|^{\kappa}}. \tag{24} \]

The result in (i) is very important as it shows that by decreasing the probability of contract execution (i.e., generating a more rare event) results in a smaller MSE. Moreover, in (ii) we show that the computational complexity of SubSim is inversely proportional to the square of the target CV $\delta^2$ and directly proportional to the execution probability $p_E$. On one hand, as the target CV becomes smaller (i.e., we demand a more accurate output), the cost increases as the method uses more subsets and subsequently a larger number of samples. On the other hand, as the execution probability decreases, the absolute value of its logarithm increases, resulting in a higher computational cost as the lower the execution probability the more demanding the estimation of $\hat{P}$ becomes. Figure 2 shows the results of a simulation run (repeated 100 times) to compare how the MSE and the computational complexity scale with respect to $p_E$ according to the SubSim theory and the experimental outputs.

[Figure 2 about here.]

4 Barrier Option Pricing using Subset Simulation Method

4.1 Geometric Brownian Motion (GBM)

The starting point in option pricing is modeling the price $S_t$ of the underlying asset. Following [Hull (2009)], assume that $S_t$ follows the SDE

\[ dS_t = S_t \mu(t) \, dt + S_t \sigma(t) \, dW_t, \tag{25} \]
a risk–neutral process, where $\mu(t)$ is the drift, $\sigma(t)$ is volatility, and $W_t$ is the standard Brownian motion. The discretized solution of (25) can then be written as follows

$$S_n = S_{n-1} \exp \left( \left( \mu_n - \frac{\sigma_n^2}{2} \right) \Delta t + \sigma_n \sqrt{\Delta t} Z_n \right),$$

(26)

where $Z_1, \ldots, Z_N \sim \mathcal{N}(0,1)$ are i.i.d. standard normal random variables.

4.2 SubSim for Barrier Options

We now consider how SubSim can be used specifically for pricing barrier options and why it is especially efficient for options on assets with high volatility. The goal is to estimate the barrier option price $P$, which is given by the following discounted expectation under the risk–neutral measure $Q$:

$$P = \mathbb{E}^Q \left[ h(S_N) \prod_{n=1}^{N} I_{[L_n,U_n]}(S_n) \right],$$

(27)

where $h(S_N)$ is the payoff at the end of the contract ($t = T$), $h(S_N) = \max\{S_N - K, 0\}$, $K$ is the strike price, and $I_{[A,B]}(x)$ stands for the indicator function: $I_{[A,B]}(x) = 1$ if $A \leq x \leq B$ and zero otherwise. Let $E \subset \mathbb{R}^N$ be a set of vectors $Z = (Z_1, \ldots, Z_N)$ that lead to a positive payoff. In other words, $E$ consists of all vectors $Z$ that result into the asset price trajectories that remain within barriers and end up above the strike price. This is schematically illustrated in Figure 3.

Let $\pi$ be the payoff function,

$$\pi(Z) = \begin{cases} S_N - K, & \text{if } Z \in E, \\ 0, & \text{if } Z \notin E. \end{cases}$$

(28)

The option price, which in our case is the expected payoff of the contract at matu-
rity, can be then rewritten as follows:

\[
P = \mathbb{E}[\pi(Z)]
= \mathbb{E}[\pi(Z) | Z \in E] \mathbb{P}(Z \in E) + \mathbb{E}[\pi(Z) | Z \notin E] \mathbb{P}(Z \notin E)
= \mathbb{E}[\pi(Z) | Z \in E] \mathbb{P}(Z \in E) = \mathbb{E}[S_N - K | Z \in E] \mathbb{P}(E)
= \mathbb{P}(E) \mathbb{E}[S_N | Z \in E] - K.
\] (29)

The first factor, the probability of positive payoff \( p_E = \mathbb{P}(E) \), can be readily estimated by SubSim,

\[
\mathbb{P}(E) \approx \hat{p}_E^{\text{SubSim}}.
\] (30)

To make SubSim directly applicable, we just need to specify the performance function \( g(S(Z)) \) that quantifies how far the asset price trajectory \( S = (S_1, \ldots, S_N) \) is from the positive payoff or, equivalently, how far \( Z = (Z_1, \ldots, Z) \) is from \( E \). We define it as follows:

\[
g(S) = \sum_{n=1}^{N} g_n(S_n),
\] (31)

where terms \( g_n(S_n) \) quantify how far the asset prices \( S_n \) is from the barriers and strike,

\[
g_n(S_n) = \begin{cases} 
U_n - S_n, & \text{if } S_n > U_n, \\
S_n - L_n, & \text{if } S_n < L_n, \\
0, & \text{otherwise.}
\end{cases}
\] (32)

\[
g_N(S_N) = \begin{cases} 
U_N - S_N, & \text{if } S_N > U_N, \\
S_N - K, & \text{if } S_N < K, \\
0, & \text{otherwise.}
\end{cases}
\]

The difference between \( g_n \) for \( n = 1, \ldots, N - 1 \) and \( g_N \) stems from the fact that at the final time \( t_N = T \), the role of the lower barrier is played by the strike price \( K \). The performance function \( g \) is schematically shown in Figure [Fig]. In terms of \( g \), the
positive-payoff event $E$ can be written as follows:

$$E = \{ Z = (Z_1, \ldots, Z_N) : g(S(Z)) \geq 0 \}.$$  \hfill (33)

Thus, the defined performance function brings the problem of estimating the probability of positive payoff $p_E$ into the general SubSim framework described in Section 2.

Moreover, the conditional expectation in (29) can be estimated using the samples generated by SubSim at the last level. Namely, let $Z^{(1)}, \ldots, Z^{(m)}$ be the last batch of MMA samples generated by SubSim before it stops,

$$Z^{(1)}, \ldots, Z^{(m)} \sim \mathcal{N}(z|E_{L-1}), \quad E_{L-1} \supset E_L \equiv E,$$  \hfill (34)

where $\mathcal{N}(z|A) \propto \mathcal{N}(z)|A(z)$ denotes the standard multivariate normal distribution conditioned on $A$. By construction (this is the SubSim stopping criterion), at least $\tilde{m} = \beta m$ of these samples are in $E$. Let

$$Z^{(1)}, \ldots, Z^{(m^*)} \sim \mathcal{N}(Z|F), \quad \beta m \leq m^* < m,$$  \hfill (35)

denote those samples. The conditional expectation can then be estimated as follows:

$$E[S_N|Z \in E] \approx \hat{E}_Q^{\text{SubSim}} = \frac{1}{m^*} \sum_{i=1}^{m^*} S_N(Z^{(i)}),$$  \hfill (36)

where $S_N(Z^{(i)}) = S_N(Z_1^{(i)}, \ldots, Z_N^{(i)})$ is the final value of the asset price obtained from (26). The expression in (36) in essence gives the expected terminal price of the underlying asset under the risk–neutral measure as the average of all the generated asset price paths. Combining (30) and (35), we obtain the SubSim estimate of the option price:

$$P \approx \hat{P}_{\text{SubSim}} = \hat{p}_E^{\text{SubSim}}(\hat{E}_Q^{\text{SubSim}} - K).$$  \hfill (37)

If the asset price $S_t$ has high volatility, then discrete asset price trajectories
$S_1, \ldots, S_N$ will have large variability and with large probability will either cross the barriers and expire or end up below the strike. This means that having a positive payoff will be a rare event. This suggests – and we confirm this by simulation in Section 5 – that SubSim should be substantially more efficient in estimating prices of barrier options on high volatility assets than MC-based methods.

5 Numerical Examples

5.1 Barrier Options

Our numerical experiments focus on pricing double knock-out barrier call options, but it is straightforward to extend the proposed methodology to other types of barrier options. Suppose that barriers are monitored during time period $[0, T]$ at equally spaced times $0 = t_0 < t_1 < \ldots < t_N = T$ with frequency $\Delta t = T/N$, and the option expires if the asset $S_t$ hits either the upper $U$ or the lower $L$ barrier. Let us denote the corresponding asset prices by $S_n = S_{t_n}$, the drift by $\mu_n = \mu(t_n)$ and the volatility by $\sigma_n = \sigma(t_n)$.

The quantity of interest is the barrier option price at the beginning of the contract ($t_0 = 0$), given by (27), which takes a non-zero value only in case the asset price trajectory remains within the two barriers. For illustrative purposes, Figure 5 shows several asset trajectories that lead to both option expiration and positive payoff.

[Figure 5 about here.]

5.2 Simulation results for SubSim vs standard MC

In the first of our numerical experiments, following Shevchenko and Del Moral (2017), we consider a double knock-out barrier call option with a starting price (spot) $S_0 = 100$, strike $K = 100$, and constant lower and upper barriers $L = 90$ and $U = 110$. A double knock-out option expires worthless in case either the upper or the lower barrier is crossed by the asset price trajectory over the life of the option, $[0, T]$. In any other case, the payoff at maturity is calculated as a plain
vanilla European call option (i.e., \( P = (S_T - K)^+ \), where \( S_T \) is the terminal asset price). The option is discretely monitored during time period \([0, T = 1]\) at equally spaced times \( 0 = t_0 < t_1 < \ldots < t_N = 1 \) with frequency \( \Delta t = T/N \), where \( N = 250 \) (approximate number of trading days in a financial year). We further assume that the drift of the underlying asset is constant, \( \mu = 0.1 \). To observe the effect of high volatility, we vary the value of \( \sigma \) over ten different values logarithmically spaced between \( \sigma_{\text{min}} = 0.2 \) and \( \sigma_{\text{max}} = 0.4 \).

The quantity of interest is the fair option price at the beginning of the contract \((t_0 = 0)\) which is given by

\[
P_0 = P \exp \left( -\int_0^T r(t) dt \right),
\]

where \( P \) is the value of the option at the end of time period given by (27) and estimated by (37), \( e^{-\int_0^T r(t) dt} \) is the discounting factor from maturity \( t_N = T \) to \( t_0 = 0 \), and \( r(t) \) is the interest rate, which is assumed to be constant in this example, \( r = 0.1 \).

First, we use SubSim with \( m = 50,000 \) samples per subset to estimate both the probability \( p_E \) of having a positive payoff at the end of the period, \( p_E \approx \hat{p}_E^{\text{SubSim}} \), and the option price,

\[
P_0 \approx \hat{P}_0^{\text{SubSim}} = \hat{P}_{\text{SubSim}} e^{-rT}.
\]

The mean values of estimates and their CVs computed from 100 independent runs of the SubSim algorithm are presented in Table 1. As expected, as the asset volatility \( \sigma \) increases, the event of having a positive payoff becomes increasingly rare (e.g. if \( \sigma = 0.4 \), then \( p_E \approx 2 \times 10^{-7} \)) and, as a result, the option becomes cheaper. The right plot in Figure 6 shows the average (based on 100 runs) total number of samples \( M \) used by SubSim versus the volatility \( \sigma \). The obtained trend is again expected: as \( \sigma \) increases, the probability \( p_E \) becomes smaller, and, therefore, the number \( L \) of subsets in (14) increases, which leads to the increase in the total number of samples (15).

The mean values of estimates and their CVs computed from 100 independent runs of the SubSim algorithm are presented in Table 1. As expected, as the asset volatility \( \sigma \) increases, the event of having a positive payoff becomes increasingly rare (e.g. if \( \sigma = 0.4 \), then \( p_E \approx 2 \times 10^{-7} \)) and, as a result, the option becomes cheaper. The right plot in Figure 6 shows the average (based on 100 runs) total number of samples \( M \) used by SubSim versus the volatility \( \sigma \). The obtained trend is again expected: as \( \sigma \) increases, the probability \( p_E \) becomes smaller, and, therefore, the number \( L \) of subsets in (14) increases, which leads to the increase in the total number of samples (15).

Next, we use MCS to estimate \( p_E \) and \( P_0 \). To ensure fair comparison of the two
methods, for each value of $\sigma$, MCS is implemented with the same total number of samples as in SubSim. The mean values of Monte Carlo estimates for the execution probability $\hat{p}_E^{MC}$ and the option price $\hat{P}_0^{MC} = \hat{P}_{MCE}^{-rT}$, with their CVs are presented in Table 1. The mean values of $\hat{p}_E^{SubSim}$ and $\hat{P}_0^{SubSim}$, which confirms that SubSim estimates are approximately unbiased. The CVs, however, differ drastically. Namely, $\delta(\hat{p}_E^{SubSim})$ and $\delta(\hat{P}_0^{SubSim})$ are substantially smaller than $\delta(\hat{p}_E^{MC})$ and $\delta(\hat{P}_0^{MC})$, respectively. This effect is more pronounced the larger the volatility. For example, if $\sigma = 0.4$, then SubSim is approximately 20 times more efficient than MCS, i.e. on average, SubSim produces 20 times more accurate estimates, where the accuracy is measured by the CV. As explained at the end of Section 2 this result stems from the fact that SubSim is more efficient than MCS in estimating small probabilities of rare events, and if volatility is large, then the event of having a positive payoff is rare.

To visualize how SubSim outperforms MCS as the volatility increases, in the left plot of Figure 6, we plot the ratios of CVs $\delta(\hat{p}_E^{MC})/\delta(\hat{p}_E^{SubSim})$ and $\delta(\hat{P}_0^{MC})/\delta(\hat{P}_0^{SubSim})$ versus $\sigma$. Since the mean values of SubSim and MCS estimates are approximately the same, the ratios of CVs are approximately the ratios of the corresponding standard errors. Graphically, the cases where SubSim outperforms MC for the estimation of the execution probability and the option price are those for which the corresponding value of $\delta(\hat{p}_E^{MC})/\delta(\hat{p}_E^{SubSim})$ or $\delta(\hat{P}_0^{MC})/\delta(\hat{P}_0^{SubSim})$ lies above the horizontal line $y = 1$ (dotted line in Figure 6). At that level, both methods would exhibit the same level of accuracy measured by the CV, since $\delta_{MC} = \delta_{SubSim}$. We notice that SubSim outperforms MC in every examined case as both lines lie above the $y = 1$ level.

[Figure 6 about here.]

In the second of our simulation tests we increase the number of samples to $m = 200,000$ using also different levels for the lower and the upper barrier. The reason we consider more samples is to compare SubSim against not only MC but also MLMC (see subsection 5.3), where $m = 200,000$ is considered in the original barrier option numerical experiments. To maintain a fair comparison we perform our MC tests with the same number of samples as in SubSim. The top graph of
Figure 7 plots the ratio of CV between SubSim and standard MC with respect to the volatility of the underlying asset for four levels of the upper and lower barrier. It is immediately noticeable that for volatility values up to 0.25 the two methods have comparable CVs (SubSim outperforms standard MC as reported in Table 3 but not significantly), providing evidence that for low–volatility assets the two methods produce sufficiently accurate results. This result is not surprising as SubSim is designed by construction to deal with problems with extremely small execution probabilities.

However, as volatility increases, SubSim outperforms naive MC in all barrier levels, while especially in the case of $L = 90$ and $U = 110$ (barriers close to $S_0$) and $\sigma \geq 0.40$ (a high–volatility asset), SubSim is up to 50 times more efficient than standard MC; for lower levels of $\sigma$, SubSim still outperforms MCS.

5.3 Simulation results for SubSim vs MLMC

In this section we compare the performance of SubSim against the MLMC method (Giles [2008b]), when both used to price a double knock–out barrier call option with two fixed barriers set at four different levels, while all the other parameters remain the same as in subsection 5.2. The original MLMC method was developed to price single knock–out barrier options, amongst other exotic derivatives, and thus we add a component for the second barrier in order to accommodate double barrier options as well (see A and B).

The price at $t = 0$ of the asset is $S_0 = 100$, the strike price is $K = 100$ and the time–increment is $\Delta t = h = T/n$ where $n$ represents the number of discrete monitoring points of the barrier option. In the case of MLMC $n$ varies between levels as it is a function of a constant $M$ and level $l$, where $l = 0, 1, 2, \ldots, L$. The barriers take four different values in increments of ten between 60 and 90 (lower) and 110 and 140 (upper). The drift of the diffusion equation is equal to $\mu = 0.10$, while the volatility (diffusion coefficient) varies between 0.05 and 0.45 taking nine

---

7To perform the simulations we use mainly the codes provided by Mike Giles at https://people.maths.ox.ac.uk/gilesm/mlmc/ doing the necessary adjustments in file mcqmc06.m.
discrete values linearly spaced in this interval. Finally, the risk–free rate at which we discount the terminal payoffs in the simulations is known and fixed at $r = 0.10$.

The bottom graph of Figure 7 plots the ratio of CV between SubSim and MLMC for four levels of barriers against asset’s volatility. For barriers which lie far from the price of the asset at $t = 0$ (i.e., [60, 140] and [70, 130] represented by the solid and the dotted line respectively), MLMC produces more accurate results than SubSim. Nevertheless, we notice that as asset volatility increases the performance of SubSim improves, approaching that of MLMC without surpassing it. SubSim outperforms MLMC when $L = 90$ and $U = 110$ (dashed/dotted line) and when $L = 80$ and $U = 120$ (dashed line) and the volatility of the underlying asset is higher than 0.25. In both cases, the probability of a non–zero payoff at $t = T$ is extremely small (Table 1), and hence the use of SubSim provides more accurate results compared to either standard MC or MLMC. The evidence we obtain here further supports the findings in Section 5.2 that SubSim is an efficient technique to price barrier options on high volatility assets, especially when the barriers are close to the initial price of the underlying asset.

Exact values for $\hat{P}_0^{\{MC,MLMC,SubSim\}}$ (option price at $t = 0$ for each of the three methods) and $CV_{\hat{P}_0^{\{MC,MLMC,SubSim\}}}$ can be found in Tables 2 and 3, respectively. For visualization purposes, we also plot these results in Figures 8 and 9.

6 Conclusion

In this paper, we develop a new stochastic simulation method for pricing barrier options. The method is based on Subset Simulation (SubSim), a very efficient algorithm for estimating small probabilities of rare events, originally developed in Au and Beck (2001) for assessing the reliability of structural engineered systems subject to uncertain excitations. The key observation allowing to exploit the efficiency of SubSim is that the barrier option price can be written as a function of the probability of option execution and a certain conditional expectation, which both can be estimated by SubSim. In the case of barrier options on high–volatility assets, the main idea of SubSim applies to the very small probability of the contract to remain
valid until maturity. We first compare the proposed SubSim method against the standard Monte Carlo simulation (MCS) to show that SubSim always outperforms MCS, confirming this with a series of numerical examples. Moreover, we show that the higher the volatility of the underlying asset (i.e., the smaller the probability of option execution), the larger the advantage of SubSim over MCS. Next, we compare our proposed method with the multilevel Monte-Carlo (MLMC) simulation introduced in [Giles (2008b)]. Although MLMC outperforms SubSim in general, we find that SubSim can still be more efficient than MLMC, – where efficiency is measured by the coefficient of variation (CV) – in cases the volatility of the underlying asset is high and the barriers are set close to the starting price of the asset. As a result, the method we propose here complements MLMC, handling special cases of barrier option settings more efficiently.

Acknowledgement: We would like to thank participants at Caltech, University of Liverpool, and Monash University seminars for helpful comments.
References


A Probability of survival of a barrier option

The pricing of barrier options is a first passage time problem in which we are interested in the first time that the price trajectory of the underlying asset crosses a prespecified barrier. Now, assuming that $U > S_0$ and $L < S_0$ are the upper and lower barriers respectively, the survival indicator function of the barrier option in (27) can be approximated via its discrete form

$$\prod_{i=0}^{n-1} I\{\hat{M}_i \leq U \land \hat{m}_i \geq L\}$$

(40)

where $\hat{M}_i$ and $\hat{m}_i$ are the maximum and minimum, respectively, of equation (26) in $[0, nh]$ and $T = nh$ or $h = T/n$ is the size of the timestep on a discrete grid. Equation (40) takes the value one if and only if the conditions for $\hat{M}_i$ and $\hat{m}_i$ are met at every time–step of the discretized problem, otherwise the product (40) becomes zero and the option expires worthless. Following Glasserman (2013) (see particularly section 6.4 and example 2.2.3) we sample the minimum and the maximum of $S$ by formulating the following problem:

$$M(t) = \max_{0 \leq u \leq t} S(u)$$

(41)

with

$$\hat{M}^h(n) = \max\{S(0), S(h), S(2h), \ldots, S(nh)\}$$

(42)

the maximum of the approximation of $S$ on $[0, nh]$, and

$$m(t) = \min_{0 \leq u \leq t} S(u)$$

(43)

with

$$\hat{m}^h(n) = \min\{S(0), S(h), S(2h), \ldots, S(nh)\}$$

(44)

the minimum of a discrete time approximation of $S$ on $[0, nh]$.

In the sampling of the maximum, conditioning on the endpoints $S(0)$ and $S(T)$, the process $\{S(t), 0 \leq t \leq T\}$ becomes a Brownian bridge, and thus we sample from the distribution of the maximum of a Brownian bridge, a Rayleigh distribution,
which results in
\[ M(T) = \frac{S(T) + \sqrt{S(T)^2 - 2T \log X}}{2}, \] (45)

where \( X \) is a uniformly distributed random variable in \([0, 1]\). Now, let \( \hat{S}_{ih} \) be a
discrete time approximation of the solution of \( S \) in (25), where \( i = 0, 1, \ldots, n, h = T/n \). To obtain a good estimation for \( \hat{M}^h \) (i.e. the maximum of the interpolating
Brownian bridge) and decrease the error induced by the discretization (i.e. the case
where \( S_u \) crosses \( U \) or \( L \) between two grid points), we interpolate over \([ih, (i+1)h]\),
which given the endpoints \( S_i \) and \( S_{i+1} \) results in
\[ M_i = \frac{S(i) + S(i+1) + \sqrt{[S(i+1) - S(i)]^2 - 2b^2h \log X}}{2} \] (46)

with \( X \sim \text{Unif}[0,1] \).

Given a barrier \( U \), the probability of survival for the option (the maximum price
of the underlying asset to remain below \( U \)) in the fine–path estimation is given by
\[ \hat{p}_{i,U} = P(\hat{M}_i \leq U|\hat{S}_i, \hat{S}_{i+1}) = 1 - \exp \left(-\frac{2(U - \hat{S}_i)(U - \hat{S}_{i+1})}{b^2h}\right), \] (47)

where \( b \) is the fixed standard deviation of the underlying asset price and \( h \) is the
time–step in the discretization process. The corresponding estimation for a coarse–
path is equal to
\[ \hat{p}_{i,U} = P(\hat{M}_i \leq U|\hat{S}_i, \hat{S}_{i+1}) = \begin{cases} 
1 - \exp \left(-\frac{2(U - \hat{S}_i)(U - \hat{S}_{i+1})}{b^2h}\right) \\
\times \left\{ 1 - \exp \left(-\frac{2(U - \hat{S}_{i+1/2})(U - \hat{S}_{i+1})}{b^2h}\right) \right\} \end{cases}. \] (48)

B Minimum of Brownian bridge

We now derive analytically the probability of survival for a double barrier option in
a fine path estimation, by calculating also the probability of the minimum of \( \hat{S} \) to
cross the lower barrier \( L \). Conditioning on endpoints \( \hat{S}_i \) and \( \hat{S}_{i+1} \), the distribution
of the minimum of the Brownian bridge (interpolated over \([i, (i+1)h]\)) is given by

\[
m_i = \frac{S(i) + S(i+1) - \sqrt{[S(i+1) - S(i)]^2 - 2b^2h \log X}}{2}, \tag{49}
\]

where \(X \sim \text{Unif}[0,1]\). Subsequently, the probability of the minimum \(m_i\) of \(\hat{S}\) to cross the lower barrier \(L\) is equal to

\[
P(m_i \leq L | \hat{S}_i, \hat{S}_{i+1})
\]

\[
\begin{aligned}
&= P\left(\frac{\hat{S}(i) + \hat{S}(i+1) - \sqrt{[\hat{S}(i+1) - \hat{S}(i)]^2 - 2b^2h \log X}}{2} \leq L | \hat{S}_i, \hat{S}_{i+1}\right) \\
&= P\left(\sqrt{[\hat{S}(i+1) - \hat{S}(i)]^2 - 2b^2h \log X} \geq (\hat{S}(i) + \hat{S}(i+1)) - 2L | \hat{S}_i, \hat{S}_{i+1}\right) \\
&\geq (\hat{S}(i) + \hat{S}(i+1))^2 - 4L(\hat{S}(i) + \hat{S}(i+1)) + 4L^2 | \hat{S}_i, \hat{S}_{i+1}\) \\
&= P\left((\hat{S}(i+1)^2 - 2\hat{S}(i)\hat{S}(i+1) + \hat{S}(i)^2 - 2b^2h \log X \\
\geq \hat{S}(i)^2 + \hat{S}(i+1)^2 + 2\hat{S}(i)\hat{S}(i+1) - 4L(\hat{S}(i) + \hat{S}(i+1)) + 4L^2 | \hat{S}_i, \hat{S}_{i+1}\right) \\
&= P\left(\log U \leq \frac{2\hat{S}(i) - 2L(\hat{S}_{i+1} - L)}{b^2h} | \hat{S}_i, \hat{S}_{i+1}\right) \\
&= P\left(\log U \leq -\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h} | \hat{S}_i, \hat{S}_{i+1}\right) \\
&= P\left(U \leq \exp\left(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\right) | \hat{S}_i, \hat{S}_{i+1}\right) = \exp\left(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\right).
\end{aligned} \tag{50}
\]

The probability in \([50]\) refers to the case of the running minimum crossing the lower barrier. The probability to remain above the lower barrier is thus equal to its complement

\[
\hat{p}_{i,L} = 1 - \exp\left(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\right), \tag{51}
\]
and the probability of the asset price to remain within the barriers on \([0, T]\) is equal to

\[
\hat{p}_i = \hat{p}_{i,U}\hat{p}_{i,L} = \left\{1 - \exp\left(-\frac{2(U - \hat{S}_i)(U - \hat{S}_{i+1})}{b^2 h}\right)\right\}\left\{1 - \exp\left(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2 h}\right)\right\}.
\] (52)

The calculation of the probability of survival for the coarse path estimation follows trivially from (52) by adjusting it using (48). Then, the option remains alive until time \(t = T = nh\) when the asset price is bounded between \(L\) and \(U\), which in the case of a coarse path estimation, using a midpoint equal to \(i + 1/2\), equals

\[
\hat{p}_i = \left\{1 - \exp\left(-\frac{2(U - \hat{S}_i)(U - \hat{S}_{i+1/2})}{b^2 h}\right)\right\}\left\{1 - \exp\left(-\frac{2(U - \hat{S}_{i+1/2})(U - \hat{S}_{i+1})}{b^2 h}\right)\right\}
\times \left\{1 - \exp\left(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1/2} - L)}{b^2 h}\right)\right\}\left\{1 - \exp\left(-\frac{2(\hat{S}_{i+1/2} - L)(\hat{S}_{i+1} - L)}{b^2 h}\right)\right\}.
\] (53)
C Numerical examples

[Figure 8 about here.]

[Figure 9 about here.]
[Table 2 about here.]

[Table 3 about here.]
Figure 1: Schematic illustration of Subset Simulation. First, MC samples $U^{(1)}, \ldots, U^{(m)}$ are generated. Next, $\tilde{m} = \beta m$ “seeds” (the closest samples to $E$) are chosen and MMA is used to generate $V^{(1)}, \ldots, V^{(m)}$ from these seeds in the direction of $E$. The SubSim algorithm proceeds in this way until the target rare event $E$ has been reached and sufficiently sampled. In this visualization, $m = 6$ and $\beta = 1/3$. 
Figure 2: Mean squared error of $\hat{P}$ with respect to the probability of execution $p_E$ on the horizontal axis (left). The simulation results show that the MSE scales like $1/|\log p_E|^k$, where $k \leq 3$, is a constant. In accordance with the theoretical findings, simulated MSE drops with increasing $p_E$. Computational cost or complexity of Subset Simulation with respect to the probability of execution (right). The simulation results show that the cost can be bounded above by a function of $|\log p_E|^r$, $r \leq 4$. 
Figure 3: Target event. The target event $E$ consists of all $Z$-vectors that lead to the positive payoff (option execution). The mapping between $Z$-space and $S$-space is given by Eq. (2).
Figure 4: Performance function. The performance function $g(S)$ quantifies how far the asset price trajectory $S$ is from the positive payoff, which occurs when $S$ stays between the barriers $U$ and $L$ (constant in this example) and ends up above the strike $K$. The value of $g(S)$ on the depicted trajectory is the negative sum of the heights of the vertical bars above the upper barrier (red), below the lower barrier (blue), and ending below the strike (purple).
Figure 5: Asset price trajectories. The top panel shows two asset trajectories that lead to a zero payoff: one trajectory breaks the upper barrier $U$ at time $t_k$, the other ends up below the strike, $S_N < K$. The bottom panel shows an asset price trajectory that results in a positive payoff $S_N - K$. For the sake of illustration, both lower and upper barriers are constant.
Figure 6: Ratios of coefficient of variation (CVs) $\delta(\hat{p}_{MC}^E)/\delta(\hat{p}_{SubSim}^E)$ and $\delta(\hat{P}_{MC}^0)/\delta(\hat{P}_{SubSim}^0)$ versus the volatility $\sigma$ (left). Total number of samples used in Subset Simulation when $L = 90$ and $U = 110$ (right).
Figure 7: Ratios of coefficient of variations (CVs) of the option price $P_0$ with respect to asset volatility, for Subset Simulation against Monte Carlo (top) and Subset Simulation against multilevel Monte Carlo (bottom). Four different barrier levels are presented.
Figure 8: Barrier option price for the three methods with respect to volatility. The four graphs correspond to different levels of the upper and lower barriers.
Figure 9: Coefficient of variation (CV) for the three methods with respect to volatility for 100 runs of the pricing algorithm. The four graphs correspond to different levels of the upper and lower barriers.
Table 1: Simulation results. This table shows the mean values and coefficients of variations (CVs) of the estimates of the execution probability $p_E$ and the barrier option price $P_0$, obtained by SubSim and MCS for different values of volatility $\sigma$. All statistics are obtained from 100 independent runs of the algorithms.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$p_{SubSim}^E / p_{MC}^E$</th>
<th>$P_{SubSim}^0 / P_{MC}^0$</th>
<th>$\delta(p_{SubSim}^E) / \delta(p_{MC}^E)$</th>
<th>$\delta(P_{SubSim}^0) / \delta(P_{MC}^0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.200</td>
<td>$8.30 \times 10^{-3}$ / $8.26 \times 10^{-3}$</td>
<td>$2.93 \times 10^{-2}$ / $2.91 \times 10^{-2}$</td>
<td>0.030 / 0.0281</td>
<td>0.034 / 0.0347</td>
</tr>
<tr>
<td>0.216</td>
<td>$4.32 \times 10^{-3}$ / $4.34 \times 10^{-3}$</td>
<td>$1.52 \times 10^{-2}$ / $1.53 \times 10^{-2}$</td>
<td>0.032 / 0.0391</td>
<td>0.036 / 0.0476</td>
</tr>
<tr>
<td>0.233</td>
<td>$2.04 \times 10^{-3}$ / $2.04 \times 10^{-3}$</td>
<td>$7.18 \times 10^{-3}$ / $7.19 \times 10^{-3}$</td>
<td>0.039 / 0.0596</td>
<td>0.044 / 0.0673</td>
</tr>
<tr>
<td>0.252</td>
<td>$8.67 \times 10^{-4}$ / $8.76 \times 10^{-4}$</td>
<td>$3.06 \times 10^{-3}$ / $3.11 \times 10^{-3}$</td>
<td>0.048 / 0.0788</td>
<td>0.055 / 0.0985</td>
</tr>
<tr>
<td>0.272</td>
<td>$3.23 \times 10^{-4}$ / $3.21 \times 10^{-4}$</td>
<td>$1.14 \times 10^{-3}$ / $1.15 \times 10^{-3}$</td>
<td>0.057 / 0.126</td>
<td>0.062 / 0.160</td>
</tr>
<tr>
<td>0.294</td>
<td>$1.06 \times 10^{-4}$ / $1.08 \times 10^{-4}$</td>
<td>$3.75 \times 10^{-4}$ / $3.80 \times 10^{-4}$</td>
<td>0.060 / 0.217</td>
<td>0.069 / 0.282</td>
</tr>
<tr>
<td>0.317</td>
<td>$2.91 \times 10^{-5}$ / $2.63 \times 10^{-5}$</td>
<td>$1.03 \times 10^{-4}$ / $9.38 \times 10^{-5}$</td>
<td>0.076 / 0.406</td>
<td>0.081 / 0.476</td>
</tr>
<tr>
<td>0.343</td>
<td>$6.85 \times 10^{-6}$ / $5.66 \times 10^{-6}$</td>
<td>$2.46 \times 10^{-5}$ / $2.14 \times 10^{-5}$</td>
<td>0.099 / 0.759</td>
<td>0.109 / 1.014</td>
</tr>
<tr>
<td>0.370</td>
<td>$1.31 \times 10^{-6}$ / $9.93 \times 10^{-7}$</td>
<td>$4.69 \times 10^{-6}$ / $3.06 \times 10^{-6}$</td>
<td>0.153 / 1.971</td>
<td>0.160 / 2.337</td>
</tr>
<tr>
<td>0.400</td>
<td>$1.99 \times 10^{-7}$ / $2.45 \times 10^{-7}$</td>
<td>$7.20 \times 10^{-7}$ / $1.10 \times 10^{-6}$</td>
<td>0.180 / 3.844</td>
<td>0.205 / 4.017</td>
</tr>
</tbody>
</table>
Barriers

<table>
<thead>
<tr>
<th>Volatility (σ)</th>
<th>Method</th>
<th>[60,140]</th>
<th>[70,130]</th>
<th>[80,120]</th>
<th>[90,110]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>Standard MC</td>
<td>9.5559</td>
<td>9.5345</td>
<td>8.3761</td>
<td>1.9009</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>9.5549</td>
<td>9.5339</td>
<td>8.3008</td>
<td>1.7882</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>9.5573</td>
<td>9.5351</td>
<td>8.3728</td>
<td>1.8997</td>
</tr>
<tr>
<td>0.10</td>
<td>Standard MC</td>
<td>9.8679</td>
<td>8.2903</td>
<td>4.5155</td>
<td>0.6617</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>9.8271</td>
<td>8.1682</td>
<td>4.3242</td>
<td>0.5941</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>9.8656</td>
<td>8.2862</td>
<td>4.5137</td>
<td>0.6615</td>
</tr>
<tr>
<td>0.15</td>
<td>Standard MC</td>
<td>8.6454</td>
<td>5.7592</td>
<td>2.3743</td>
<td>0.1712</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>8.4688</td>
<td>5.5283</td>
<td>2.1859</td>
<td>0.1956</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>8.6413</td>
<td>5.7570</td>
<td>2.3734</td>
<td>0.1711</td>
</tr>
<tr>
<td>0.20</td>
<td>Standard MC</td>
<td>6.6578</td>
<td>3.8014</td>
<td>1.2839</td>
<td>0.0290</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>6.3772</td>
<td>3.5392</td>
<td>1.1595</td>
<td>0.0716</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>6.6477</td>
<td>3.7958</td>
<td>1.2839</td>
<td>0.0292</td>
</tr>
<tr>
<td>0.25</td>
<td>Standard MC</td>
<td>4.8993</td>
<td>2.5194</td>
<td>0.6712</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>4.5896</td>
<td>2.2841</td>
<td>0.6406</td>
<td>0.0273</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>4.8970</td>
<td>2.5148</td>
<td>0.6707</td>
<td>0.0033</td>
</tr>
<tr>
<td>0.30</td>
<td>Standard MC</td>
<td>3.5877</td>
<td>1.6833</td>
<td>0.3226</td>
<td>0.0003</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>3.2844</td>
<td>1.5152</td>
<td>0.3668</td>
<td>0.0120</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>3.5840</td>
<td>1.6792</td>
<td>0.3223</td>
<td>0.0003</td>
</tr>
<tr>
<td>0.35</td>
<td>Standard MC</td>
<td>2.6423</td>
<td>1.1106</td>
<td>0.1406</td>
<td>1.33E-05</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>2.3811</td>
<td>1.0275</td>
<td>0.2233</td>
<td>5.70E-03</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>2.6414</td>
<td>1.1096</td>
<td>0.1403</td>
<td>1.61E-05</td>
</tr>
<tr>
<td>0.40</td>
<td>Standard MC</td>
<td>1.9638</td>
<td>0.7114</td>
<td>0.0554</td>
<td>1.84E-06</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>1.7620</td>
<td>0.7101</td>
<td>0.1306</td>
<td>3.00E-03</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>1.9604</td>
<td>0.7107</td>
<td>0.0554</td>
<td>7.19E-07</td>
</tr>
<tr>
<td>0.45</td>
<td>Standard MC</td>
<td>1.4525</td>
<td>0.4387</td>
<td>0.0199</td>
<td>5.79E-08</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>1.3312</td>
<td>0.4956</td>
<td>0.0776</td>
<td>1.70E-03</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>1.4501</td>
<td>0.4371</td>
<td>0.0198</td>
<td>2.49E-08</td>
</tr>
</tbody>
</table>

Table 2: Barrier option price as derived by the three methods under comparison for four barrier levels against volatility.
## Table 3: Coefficient of variation (CV) of the three methods under comparison for four barrier levels against asset’s volatility.

<table>
<thead>
<tr>
<th>Volatility (σ)</th>
<th>Method</th>
<th>[60,140]</th>
<th>[70,130]</th>
<th>[80,120]</th>
<th>[90,110]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>Standard MC</td>
<td>0.0018</td>
<td>0.0016</td>
<td>0.0019</td>
<td>0.0045</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0004</td>
<td>0.0004</td>
<td>0.0005</td>
<td>0.0024</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0011</td>
<td>0.0013</td>
<td>0.0013</td>
<td>0.0031</td>
</tr>
<tr>
<td>0.10</td>
<td>Standard MC</td>
<td>0.0027</td>
<td>0.0026</td>
<td>0.0038</td>
<td>0.0080</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0004</td>
<td>0.0006</td>
<td>0.0010</td>
<td>0.0077</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0018</td>
<td>0.0019</td>
<td>0.0025</td>
<td>0.0059</td>
</tr>
<tr>
<td>0.15</td>
<td>Standard MC</td>
<td>0.0037</td>
<td>0.0040</td>
<td>0.0054</td>
<td>0.0177</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0005</td>
<td>0.0008</td>
<td>0.0017</td>
<td>0.0229</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0027</td>
<td>0.0026</td>
<td>0.0037</td>
<td>0.0092</td>
</tr>
<tr>
<td>0.20</td>
<td>Standard MC</td>
<td>0.0041</td>
<td>0.0053</td>
<td>0.0084</td>
<td>0.0444</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0007</td>
<td>0.0013</td>
<td>0.0044</td>
<td>0.0598</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0032</td>
<td>0.0039</td>
<td>0.0055</td>
<td>0.0156</td>
</tr>
<tr>
<td>0.25</td>
<td>Standard MC</td>
<td>0.0054</td>
<td>0.0066</td>
<td>0.0095</td>
<td>0.1122</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0009</td>
<td>0.0020</td>
<td>0.0063</td>
<td>0.1623</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0042</td>
<td>0.0053</td>
<td>0.0068</td>
<td>0.0219</td>
</tr>
<tr>
<td>0.30</td>
<td>Standard MC</td>
<td>0.0069</td>
<td>0.0089</td>
<td>0.0180</td>
<td>0.4069</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0014</td>
<td>0.0053</td>
<td>0.0104</td>
<td>0.1992</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0043</td>
<td>0.0061</td>
<td>0.0093</td>
<td>0.0347</td>
</tr>
<tr>
<td>0.35</td>
<td>Standard MC</td>
<td>0.0075</td>
<td>0.0099</td>
<td>0.0301</td>
<td>1.9758</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0020</td>
<td>0.0041</td>
<td>0.0310</td>
<td>0.2169</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0061</td>
<td>0.0072</td>
<td>0.0129</td>
<td>0.0652</td>
</tr>
<tr>
<td>0.40</td>
<td>Standard MC</td>
<td>0.0098</td>
<td>0.0126</td>
<td>0.0373</td>
<td>5.6981</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0025</td>
<td>0.0057</td>
<td>0.0288</td>
<td>0.2257</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0067</td>
<td>0.0088</td>
<td>0.0166</td>
<td>0.1047</td>
</tr>
<tr>
<td>0.45</td>
<td>Standard MC</td>
<td>0.0087</td>
<td>0.0106</td>
<td>0.0254</td>
<td>8.2893</td>
</tr>
<tr>
<td></td>
<td>MLMC</td>
<td>0.0059</td>
<td>0.0164</td>
<td>0.0538</td>
<td>0.2465</td>
</tr>
<tr>
<td></td>
<td>SubSim</td>
<td>0.0077</td>
<td>0.0128</td>
<td>0.0217</td>
<td>0.1808</td>
</tr>
</tbody>
</table>