European Journal of Operational Research 290 (2021) 313–330

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Interfaces with Other Disciplines

Pricing discretely-monitored double barrier options with small probabilities of execution☆

Vasileios E. Kontosakos[a], Keegan Mendonca[b], Athanasios A. Pantelous[c,*], Konstantin M. Zuev[d]

[a] Department of Econometrics and Business Statistics, Monash University, Wellington Rd, Clayton, Victoria 3800, Australia
[b] Department of Computing and Mathematical Sciences, California Institute of Technology, E. California Blvd., Pasadena, CA 91125, USA
[c] Department of Econometrics and Business Statistics, Monash University, Wellington Rd, Clayton, Victoria 3800, Australia
[d] Department of Computing and Mathematical Sciences, California Institute of Technology, E. California Blvd., Pasadena, CA 91125, USA

ARTICLE INFO

Article history:
Received 26 September 2019
Accepted 21 July 2020
Available online 26 July 2020

JEL classification:
G13
C15

Keywords:
Simulation
Barrier options pricing
Rare event
Path–dependent derivatives
Discrete monitoring

ABSTRACT

In this paper, we propose a new stochastic simulation-based methodology for pricing discretely-monitored double barrier options and estimating the corresponding probabilities of execution. We develop our framework by employing a versatile tool for the estimation of rare event probabilities known as subset simulation algorithm. In this regard, considering plausible dynamics for the price evolution of the underlying asset, we are able to compare and demonstrate clearly that our treatment always outperforms the standard Monte Carlo approach and becomes substantially more efficient (measured in terms of the sample coefficient of variation) when the underlying asset has high volatility and the barriers are set close to the spot price of the underlying asset. In addition, we test and report that our approach performs better when it is compared to the multilevel Monte Carlo method for special cases of barrier options and underlying assets that make the pricing problem a rare event estimation. These theoretical findings are confirmed by numerous simulation results.

1. Introduction

A barrier option is among the most actively-traded path-dependent financial derivatives whose payoff depends on whether the underlying asset[1] has reached (or exceeded) a predetermined price during the option's contract term (e.g., Dadachanji, 2015; Hull, 2009). In the financial industry, a barrier option is traded, because it more accurately represents investor's beliefs[2] and offers a more attractive risk–reward relation than the corresponding plain–vanilla option.[3] Further, a barrier option's advantage stems from its 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.[4]

In practice, to have a double barrier option with a very small probability of execution (i.e., to be a very cheap option), we should either set the barriers close to the spot price(s) of the underlying asset(s) at contract initiation or the underlying asset(s) need to be of high volatility. Apparently, in both ways, the execution of the

☆ We are extremely grateful to the three anonymous reviewers and handling editor Emanuele Borgonovo who have afforded us considerable assistance in enhancing both the quality of the findings and the clarity of their presentation. The authors would like to thank Siu-Kui (Ivan) Au, James Beck, Damiano Brigo, Gianluca Fusai, Otto Konstadatos, Steven Kou, Ioannis Kyriakou, Zili Zhu, and the participants at the Global Finance 2018, Quantitative Methods in Finance 2018 and Quantitative Finance and Risk Analysis 2019 Conferences and seminars at Caltech, University of Liverpool, Monash University and Shanghai University for helpful comments. Any remaining errors are ours.

* Corresponding author.
E-mail addresses: Vasileios.Kontosakos@monash.edu (V.E. Kontosakos), mendoncakeegan@gmail.com (K. Mendonca), Athanasios.Pantelous@monash.edu (A.A. Pantelous), kostia@caltech.edu (K.M. Zuev).

[1] In practice, mostly currencies, commodities and interest rates are used as the underlying asset(s).

[2] Actually, a down-and-out barrier call option can serve the same purpose as a plain–vanilla call, but at a much lower cost given the investor has a strong indication that the price of the underlying asset will increase. The downside of this strategy is the incomplete hedge it provides in case the market moves to the opposite direction.

[3] Note that the payoff at maturity of a barrier option is identical to that of a plain–vanilla European option, in case the price of the underlying asset has remained above the barrier (for a knock–out barrier option) or zero otherwise.

[4] Barrier options are cheaper than the corresponding plain vanilla ones because they expire more easily and are less likely to be executed (Jewitt, 2015). Further discussion about ins and outs of barrier options can be found in Derman and Kani (1996, 1997).

https://doi.org/10.1016/j.ejor.2020.07.044
0377-2217/

option will be a *rare event*.[5] In this paper, given that barrier options are usually *over-the-counter* (OTC)–traded instruments, the development of a framework able to deal efficiently with them, and in particular, when a high volatility underlying asset(s) is considered, tackles an actual and challenging problem in computational finance, which to our knowledge has not been explicitly studied so far.

The main contributions of this paper can be summarized as follows. First, we develop a novel stochastic simulation-based methodology for pricing discretely-monitored double barrier options which is based on the *subset simulation* (SubSim) method, a *Markov Chain Monte Carlo* (MCMC)–based algorithm originally introduced by Au and Beck (2001) to deal with complex engineered systems.[6]

Second, we calculate the fair price for discretely-monitored double barrier options on high volatility asset, and when, the barriers set near the starting price of the underlying asset. Further, we show that the proposed methodology is insensitive to the choice of the underlying asset(s) dynamics. However, for illustration purposes in our extensive simulation study, we consider two processes for the evolution of the asset price: first, a standard *geometric Brownian motion* (GBM) and second, the double exponential jump diffusion process proposed in Kou (2002). Under this challenging setting, the very small exercise (rare event) probability corresponds to the probability of the barrier option to be executed at maturity (i.e., the price of the underlying asset to remain within the barriers). This setting in a simple *Monte Carlo simulation* (MCS) setup results – with an extremely large probability – in asset price trajectories which cross the barriers, rendering the barrier option invalid before maturity.

Third, 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 $O(|\log p_E|^{r/2})$, where p_E is the execution probability and $2 \leq r \leq 3$ is a constant. Comparing this against the MCS estimator whose CV is $O(p_E^{-1/2})$ and for very small values of p_E, we verify in a straightforward manner that the latter increases at a dramatically faster pace compared to the SubSim estimator. Moreover, the MSE of the derived SubSim estimator is $O(p_E^2 |\log p_E|^{r-1} m^{-1})$ which is decreasing with p_E. We finally show that the computational cost (i.e., the complexity) of the SubSim estimate for the discretely-monitored double barrier option price can be bounded above by an amount proportional to $|\log p_E| m h_s^{-1}$, where m is the number of samples per subset and h_s is the time step of the discretized underlying stochastic process.

Lastly, we compare our results against another popular simulation-based methodology, namely the *multi–level Monte Carlo* (MLMC) of Giles (2008a,b) approach and show that for very small values of p_E, the SubSim estimator outperforms the MLMC estimator in terms of the observed CV. To conclude, our method might be seen as a reliable alternative to price path–dependent options that complements MLMC for special cases of underlying assets and barrier setups.

Our paper is organized as follows. In Section 2, the connection with the existing literature on barrier options is provided. In Section 3, we show how SubSim can be used specifically for the estimation of the execution probability and the option payoff at maturity. Section 4 subsequently presents the main theorem and its proof. This establishes the limiting behaviour of the MSE and the computational complexity for a broad category of applications. Finally, numerical results and comparisons with the standard MCS and the MLMC methods are presented in Sections 5 and 6 to provide support for the theoretical analysis followed by some concluding remarks and directions for further research in Section 7.

2. Literature review

As we have seen before, a barrier option is typically classified as either *knock -in* or *-out* depending on whether it is activated or expires worthless when the price of the underlying asset crosses a certain level (i.e., the barrier) (Derman & Kani, 1996; 1997; Guardasoni & Sanfelici, 2016). Barrier options were estimated that they accounted for approximately half the volume of all traded exotic options (Luenberger & Luenberger, 1999). Despite the 2007–08 credit crunch and the subsequent drop in the demand for path–dependent instruments, barrier options can still be a useful investment or hedging vehicle.[7]

Overall, the pricing of barrier options is a very challenging problem due to the need to monitor the price of the underlying asset and compare it against the barriers at multiple discrete points during the contract life (Kou, 2007). 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 & Recchioni, 2007).

Computationally, certain barrier options, such as down-and-out options, can be priced by appropriately adjusting *Black–Scholes* (BS) equation (see Lo, Lee, & Hui, 2003; Merton, 1973). This idea can be further extended to more complicated barrier options, which can be priced using replicating portfolios of vanilla options in a BS framework Carr and Chou. All these approaches, however, suffer from the BS model's dependence on a number of assumptions which are not met in real–world trading (e.g., Hull, 2009). As a result, the estimates we obtain for option prices under the *equivalent martingale measure* are often inaccurate. In addition, we find models for barrier options with analytical solutions, such as jump-diffusion models (Kou, 2002; Kou & Wang, 2004), the *constant elasticity of variance* (CEV) model (Boyle & Tian, 1999; Davydov & Linetsky, 2001), exact analytical approaches (Fusai, Abrahams, & Sgarra, 2006) and methods based on Lévy processes that use the Hilbert transform (Feng & Linetsky, 2008; Fusai, Germano, & Marazzina, 2016; Jeannin & Pistorius, 2010; Lian, Zhu, Elliott, & Cui, 2017; Phelan, Marazzina, Fusai, & Germano, 2019).

Another set of methods for pricing barrier options based on solving *partial differential equations* (PDEs) was proposed in Boyle and Tian (1998), Zvan, Vetzal, and Forsyth (2000), Buchen and Konstandatos (2009), Zhu and De Hoog (2010) and Golbabai, Ballestra, and Ahmadian (2014). 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. Other approaches used in the pricing of exotic derivatives include the method of lines (Chiarella, Kang, & Meyer, 2012), where the Greeks are also estimated, robust optimization techniques (Bandi & Bertsimas, 2014), applicable also to American options, finite–difference based approaches (Wade, Khaliq, Yousuf, Vigo-Aguiar, & Deininger, 2007), where a Crank–Nicolson smoothing strategy to treat discontinuities in barrier options is presented, and regime–switching models (Elliott, Siu, & Chan, 2014; Rambeerich & Pantelous, 2016).

[5] We should emphasise here that a barrier option on high volatility underlying asset(s) can be used in a similar way as a cheap, deep out–of–the–money option, serving as a hedge to provide insurance in a financial turmoil, given their volatility–dependence Carr and Chou. Moreover, according to Andersen, Bollerslev, Diebold, and Ebens (2001), the mean realized annual volatility of the thirty stocks in the *Dow Jones Industrial Average* (DJIA) is approximately 28% (ranging between 22% and 42%) while it is not uncommon to record stocks with volatility levels between 33% and 40%.

[6] This is later extended by Zuev, Wu, and Beck (2015) to complex networks (see, also in Au & Wang, 2014).

[7] For an up-to-date estimate of their traded volume the interested reader is referred to the OTC derivatives statistics for 2018 by the Bank of International Settlements, found here: https://www.bis.org/statistics/derstats.htm

MCS is often used for option pricing (Schoutens & Symens, 2003) and particularly for barrier options (Glasserman & Staum, 2001). In the latter, an *importance sampling* (IS) based method is developed for the pricing of several types of barrier options. Although IS is known to be an efficient algorithm for sampling, in a rare event estimation context, it is not as efficient as SubSim when sampling is performed in high-dimensional spaces, i.e., the dimension N of the approximated equation is very large (see, Au and Beck, 2003; Katafygiotis and Zuev, 2008, for the details). The main reason for the exponential drop of efficiency of the standard IS in high-dimensional spaces is that the importance sampling density, employed within IS, should be very carefully tailored to the problem under consideration, which has been proved to be a very difficult task (Beskos, Crisan, & Jasra, 2014; Snyder, Bengtsson, Bickel, & Anderson, 2008). The theoretical aspects of this phenomenon are described in (Bengtsson, Bickel, & Li, 2008). More recently the "collapse of IS in high dimensions" is discussed in Agapiou, Papaspiliopoulos, Sanz-Alonso, and Stuart (2017).

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 discretely monitored barrier option converges to that of a continuously monitored one (Broadie, Glasserman, & Kou, 1997; Phelan, Marazzina, Fusai, & Germano, 2018). However, MCS has a serious drawback, as high volatility makes it difficult for the asset to remain within barriers -especially when the gap between them is small- which in turn, makes a positive payoff a rare event (Glasserman, Heidelberger, Shahabuddin, & Zajic, 1999). As a result, any standard MCS method will be inaccurate and highly unstable (Geman & Yor, 1996). This motivates the development of more advanced stochastic simulation methods which inherit the robustness of MCS, and yet are more efficient in estimating barrier option prices. A range of stochastic simulation techniques for speeding up the convergence have been proposed, such as the MCS approximation correction for constant single barrier options (Beaglehole, Dybvig, & Zhou, 1997), the simulation method based on the large deviations theory (Baldi, Caramellino, & Iovino, 1999) and the sequential MCS method (Shevchenko & Del Moral, 2017).

3. Barrier option pricing with subset simulation

The starting point in option pricing is modeling the price S_t of the underlying asset. In our study, we consider two dynamics for the evolution of asset price; we assume that the underlying asset price follows either a GBM or a more complex double exponential jump diffusion process with i.i.d. price jumps (Kou, 2002; Kou & Wang, 2004). Although both models are well established and widely used in academic research, particularly for option pricing, we present briefly here their basic features.

3.1. GBM

Under a standard GBM, the price S_t of the underlying asset follows the *stochastic differential equation* (SDE)

$$dS_t = S_t\mu_t dt + S_t\sigma_t dW_t, \tag{1}$$

a risk–neutral process, where μ_t is the drift, σ_t is volatility, and W_t is the standard Brownian motion defined on a common probability space $(\Omega,\mathcal{F},\mathbb{P})$. The discretized solution of Eq. (1) can then be written as follows

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

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

3.2. Double exponential jump diffusion

The main contribution of Kou's jump diffusion model is the incorporation of a discrete jump component into the continuous part of the classic diffusion model as presented on the RHS of Eq. (1). The price S_t then evolves according to the following SDE:

$$dS_t = S_t\mu_t dt + S_t\sigma_t dW_t + d\Big(\sum_{i=1}^{N_t}(V_i-1)\Big), \tag{3}$$

where W_t is the standard Brownian motion as in Eq. (1) defined on the same probability space, N_t is a Poisson process with rate λ (i.e., the expected number of jumps) and $\{V_i\}$ is a sequence of i.i.d. random variables which satisfies that $X=\log V$ has an asymmetric double exponential distribution with *probability density function* (pdf) given by

$$f_X(x) = p\eta_1 e^{-\eta_1 x}1_{\mathbb{R}_+}(x) + q\eta_2 e^{\eta_2 x}1_{\mathbb{R}_-}(x). \tag{4}$$

In Eq. (4), $\eta_1>1$, $\eta_2>0$, and they control the jump magnitude, $p+q=1$ with $p,q\geq 0$ and they represent the probability for an upward and a downward jump, respectively. It follows that,

$$X=\log V\overset{d}{=}\begin{Bmatrix}\xi^+ & ,\text{with probability } p\\ -\xi^- & ,\text{with probability } q\end{Bmatrix}, \tag{5}$$

where ξ^+ and ξ^- are exponentially distributed random variables with mean values $1/\eta_1$ and $1/\eta_2$, respectively, while the $\overset{d}{=}$ symbol denotes equality in distribution. The solution to the SDE in Eq. (3), in its discretized form, yields the following asset price dynamics:

$$S_n = S_{n-1}\exp\left(\left(\mu_n-\frac{\sigma_n^2}{2}\right)\Delta t+\sigma_n\sqrt{\Delta t}Z_n\right)\prod_{i=1}^{N_n}V_i, \tag{6}$$

where again $Z_1,\dots,Z_N\sim\mathcal{N}(0,1)$ are i.i.d. standard normal random variables. For simplicity, both in the GBM case in Eq. (1) and in the jump diffusion model in Eq. (3), $\mu(t)$ and $\sigma(t)$ are time invariant and are set equal to μ and σ. In practice, our pricing framework can be applied potentially to any plausible underlying price dynamics; alternatively, we could use the constant elasticity of variance (CEV) process in place of the double exponential jump diffusion process. Notwithstanding the existence of analytical or semi-analytical solutions for the pricing of path-dependent securities (see Davydov and Linetsky, 2001; Sesana, Marazzina, and Fusai, 2014, for applications of the CEV model in pricing exotics) under the CEV process, its underlying return distribution exhibits a thinner tail compared to the jump diffusion model. As a result it fails to capture the leptokurtic feature of asset returns to the same extent as the latter (Kou, 2002).

3.3. Subset simulation for barrier options

We first consider how SubSim can be used specifically for pricing barrier options and why it is especially efficient for options with small probability of execution. The goal is to estimate the price P at $t=0$ given by the following discounted expectation under the risk–neutral measure $\mathbb{Q}$:

$$P=\mathbb{E}\left[h(S_N)\prod_{n=1}^{N}1_{[L_n,U_n]}(S_n)\right], \tag{7}$$

where $h(S_N)$ is the payoff at the contract maturity ($t=T$), $h(S_N)=\max\{S_N-K,0\}$, K is the strike price, and $1_{[A,B]}(x)$ stands for the indicator function: $1_{[A,B]}(x)=1$ if $A\leq x\leq B$, where A and B are the upper and lower barriers respectively, and zero otherwise.

In order to use the SubSim method, we need to bring the problem in Eq. (7) in a form suitable to become an input for the method. Suppose that the time–evolution of the dynamic system

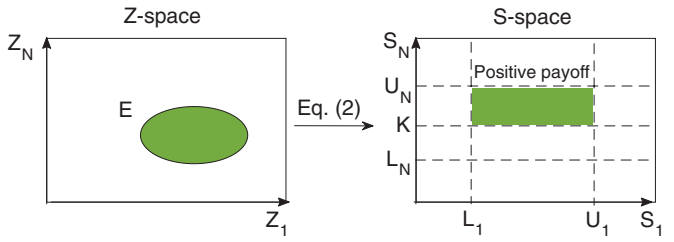

Fig. 1. Target event. The target event E consists of all Z-vectors that lead to the positive payoff (option execution). The mapping between Z- and S-spaces is given by either Eq. (2) or Eq. (6).

under study (e.g., evolution of the asset price S_n) is modeled by the following discrete model:

$$S_n = F(S_{n-1}, U_n), \quad n = 1, 2, \ldots, N, \tag{8}$$

where S_n is the price of the underlying asset at time t_n, $S = (S_1, \ldots, S_N)$ is the trajectory of the underlying asset, U_n is a random input at time t_n, and F is a certain function that governs the evolution of S (i.e., the GBM Eq. (1) or the jump diffusion in Eq. (3) in our case). Let $g(S)$ be the performance function – a function related to the quantity of interest S – (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 $g(S)$ exceeds a critical threshold α:

$$E = \{U = (U_1, \ldots, U_N) \ : \ g(S(U)) \geq \alpha\} \subset \mathbb{R}^N. \tag{9}$$

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,

$$\mathbb{R}^N = E_0 \supset E_1 \supset \ldots \supset E_L \equiv E. \tag{10}$$

The intermediate events E_i can be defined by simply repeatedly relaxing the value of the critical threshold α in Eq. (9),

$$E_i = \{U = (U_1, \ldots, U_N) \ : \ g(S(U)) \geq \alpha_i\}, \quad \alpha_1 < \alpha_2 < \ldots < \alpha_L \equiv \alpha. \tag{11}$$

To make SubSim directly applicable, we need to specify suitable functions for the underlying asset price trajectory and the expected payoff at maturity. 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 represents the target event for our problem and consists of all vectors Z that result into those asset price trajectories that remain within barriers and end up above the strike price. This is schematically illustrated in Fig. 1. Let π be the payoff function,

$$\pi(Z) = \begin{cases} S_N - K, & \text{if } Z \in E, \\ 0, & \text{if } Z \notin E, \end{cases} \tag{12}$$

equal to the payoff of a plain vanilla call in case the asset price trajectory remains within the barriers and ends up above the strike price or zero otherwise.

As for the performance function, in the case of option pricing, this quantifies how far the asset price trajectory $S = (S_1, \ldots, S_N)$ lies from the positive payoff, or equivalently, how far $Z = (Z_1, \ldots, Z)$ is from E. We define it as follows:

$$g(S) = \sum_{n=1}^{N} g_n(S_n), \tag{13}$$

where $g_n(S_n)$ quantify how far the asset prices S_n is from the barriers L_n, U_n and strike K,

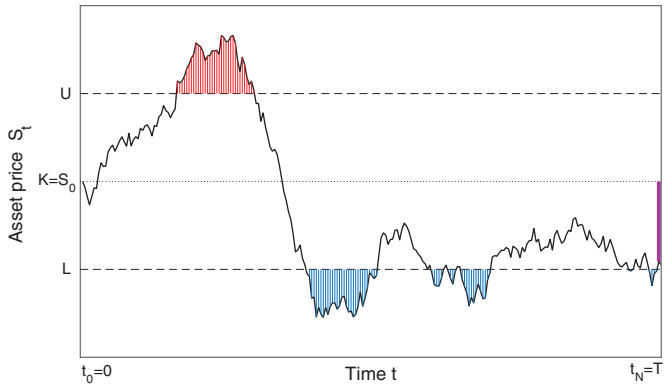

Fig. 2. Performance function. The function $g(S)$ quantifies how far the asset price trajectory S is from the positive payoff, which occurs when S stays between the barriers U and L and ends up above the strike K. The value of $g(S)$ on the depicted trajectory is the negative sum of the heights of the vertical bars above the upper barrier (red), below the lower barrier (blue), and ending below the strike (purple)..

$$g_n(S_n) = \begin{cases} U_n - S_n, & \text{if } S_n > U_n, \\ S_n - L_n, & \text{if } S_n < L_n, \\ 0, & \text{otherwise.} \end{cases} \quad \text{for } n = 1, \ldots, N-1.$$

$$g_N(S_N) = \begin{cases} U_N - S_N, & \text{if } S_N > U_N, \\ S_N - K, & \text{if } S_n < K, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The difference between g_n for $n = 1, 2, \ldots, N-1$ and g_N stems from the fact that at maturity $t_N = T$, the role of the lower barrier is played by the strike price K. The performance function g is schematically shown in Fig. 2. In terms of g, the positive-payoff event E can be written, according to the definition of the performance function $g(S)$ in Eq. (14), as follows:

$$E = \{Z = (Z_1, \ldots, Z_N) \ : \ g(S(Z)) \geq 0\}, \tag{15}$$

where α is now replaced by zero and the defined performance function brings the problem of estimating the probability of positive payoff p_E into the general SubSim framework developed in Au and Beck (2001). Then, combining Eqs. (12) and (14), the option price, which in our case is the expected payoff of the contract at maturity, can be rewritten as follows:

$$\begin{aligned} P &= \mathbb{E}[\pi(Z)] \\ &= \mathbb{E}[\pi(Z)|Z \in E]\mathbb{P}(Z \in E) + \mathbb{E}[\pi(Z)|Z \notin E]\mathbb{P}(Z \notin E) \\ &= \mathbb{E}[\pi(Z)|Z \in E]\mathbb{P}(Z \in E) = \mathbb{E}[S_N - K|Z \in E]\mathbb{P}(E) \\ &= \mathbb{P}(E)(\mathbb{E}[S_N|Z \in E] - K). \end{aligned} \tag{16}$$

Now, the problem boils down to estimating the execution probability, $p_E = \mathbb{P}(E)$, and the expectation of the payoff at maturity, given by the second term in the product of Eq. (16). The analysis carried out in the next section is insensitive to the choice of the underlying price dynamics; the results hold regardless of whether the GBM or the jump diffusion or another stochastic process is used.

4. Probability of contract execution p_E and option payoff h via SubSim

We start with the calculation of p_E to notice that given the sequence in Eq. (10), the small probability p_E of rare event E can be written as a product of conditional probabilities:

$$\begin{aligned} p_E &= \mathbb{P}(E_L) = \mathbb{P}(E_L|E_{L-1})\mathbb{P}(E_{L-1}) \\ &= \mathbb{P}(E_L|E_{L-1})\mathbb{P}(E_{L-1}|E_{L-2})\mathbb{P}(E_{L-2}) = \ldots = \prod_{i=1}^{L} \mathbb{P}(E_i|E_{i-1}). \end{aligned} \tag{17}$$

By choosing the intermediate thresholds, α_i, appropriately (in the actual implementation of SubSim described below, α_i are chosen adaptively on the fly), we can make all conditional probabilities $\mathbb{P}(E_i|E_{i-1})$ sufficiently large, and estimate them efficiently by MC-like simulation methods. In fact, the first factor in the right-hand side of Eq. (17), $\mathbb{P}(E_1|E_0)=\mathbb{P}(E_1)$, can be directly estimated by MCS:

$$\mathbb{P}(E_1)\approx\frac{1}{m}\sum_{i=1}^{m}1_{E_1}\big(U^{(i)}\big),\quad U^{(1)},\dots,U^{(m)}\sim f_U. \tag{18}$$

However, estimating the remaining factors, $\mathbb{P}(E_i|E_{i-1})$, for $i\geq 2$, is not trivial, 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 computationally demanding process, 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 & Beck, 2001; Zuev & Katafygiotis, 2011), which belongs to a large family of MCMC algorithms (Liu, 2001; Robert & Casella, 2004) for sampling from complex probability distributions.[8]

To sample from $f_U(u|E_{i-1})$, MMA generates a Markov chain whose stationary distribution is $f_U(u|E_{i-1})$. The key difference between MMA and the original Metropolis algorithm is how the "*candidate*" state of a Markov chain is generated (in Appendix A, the MMA algorithm used for the sampling is presented). Then, using the detailed balance equation, it can be shown (see Au and Beck, 2001, for the details) that if $U^{(j)}$ is distributed according to the target distribution, $U^{(j)}\sim f_U(u|E_{i-1})$, 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. Now, to estimate the small probability of execution, p_E, the method starts by generating m MCS samples, $U^{(1)},\dots,U^{(m)}\sim f_U$, and computing the corresponding system trajectories, $S^{(1)},\dots,S^{(m)}$, via Eq. (8) and performance values, $g_U^{(i)}=g(S^{(i)})$. Without loss of generality, we can assume that

$$g_U^{(1)}\geq g_U^{(2)}\geq\dots\geq g_U^{(m)}. \tag{19}$$

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 in Eq. (19) 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 us define the first intermediate threshold, α_1, as the average between the performance values of the $\tilde{m}$th and $(\tilde{m}+1)$th system trajectories, where $\tilde{m}=\beta m$ with $\beta\in(0,1)$:[9]

$$\alpha_1=\frac{g_U^{(\beta m)}+g_U^{(\beta m+1)}}{2},\qquad 0<\beta<1. \tag{20}$$

Setting α_1 to this value has two important corollaries: (1) the MCS estimate of $\mathbb{P}(E_1)$ given by Eq. (18) is exactly β, and (2) samples $U^{(1)},\dots,U^{(\beta m)}$ 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 $U^{(1)},\dots,U^{(\beta m)}$ as "*seeds*":

$$U^{(i)}=V^{(i,1)}\xrightarrow{\text{MMA}}V^{(i,2)}\xrightarrow{\text{MMA}}\dots\xrightarrow{\text{MMA}}V^{(i,l)}. \tag{21}$$

Since by construction, all seeds are in the stationary state, $U^{(i)}\sim f_U(u|E_1)$, $i=1\dots,\tilde{m}$, so are all Markov chains states, $V^{(i,j)}\sim f_U(u|E_1)$, $j=1,2,\dots,l$. 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 us denote samples $V^{(i,j)}$ by simply $V^{(1)},\dots,V^{(m)}$. Next, the second intermediate threshold, α_2, is similarly defined as follows:

$$\alpha_2=\frac{g_V^{(\beta m)}+g_V^{(\beta m+1)}}{2}, \tag{22}$$

where $g_V^{(1)}\geq g_V^{(2)}\geq\dots\geq g_V^{(m)}$ are the ordered performance values corresponding to samples $V^{(1)},\dots,V^{(m)}$. Again, by construction, $\mathbb{P}(E_2|E_1)\approx\beta$ and $V^{(1)},\dots,V^{(\beta m)}\sim f_U(u|E_2)$. The SubSim method, schematically illustrated in Fig. 3, 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 Eq. (17) are then approximated by β and $\mathbb{P}(E|E_{L-1})\approx m_E/m$. This results into the following estimate:

$$p_E\approx\hat{p}_E^{\text{SubSim}}=\beta^{L-1}\frac{m_E}{m}, \tag{23}$$

where L[10] is the number of subsets in Eq. (17) required to reach E. The total number of samples used by SubSim is then

$$M=\underbrace{m}_{\text{MCS}}+\underbrace{m(1-\beta)(L-1)}_{\text{MMA}}. \tag{24}$$

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

$$\mathbb{P}(E)\approx\hat{p}_E^{\text{SubSim}}. \tag{25}$$

Moreover, the conditional expectation in Eq. (16) for the terminal asset price can be estimated using the samples generated by SubSim at the last level. Namely, let $Z^{(1)},\dots,Z^{(m)}$ be the last batch of MMA samples generated by SubSim before it stops,

$$Z^{(1)},\dots,Z^{(m)}\sim\mathcal{N}(z|E_{L-1}),\quad E_{L-1}\supset E_L\equiv E, \tag{26}$$

where $\mathcal{N}(z|A)\propto\mathcal{N}(z)I_A(z)$ 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

$$Z^{(1)},\dots,Z^{(m^*)}\sim\mathcal{N}(Z|E),\qquad \beta m\leq m^*<m, \tag{27}$$

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

$$\mathbb{E}[S_N|Z\in E]\approx\widehat{\mathbb{E}}_{\mathbb{Q}}^{\text{SubSim}}=\frac{1}{m^*}\sum_{i=1}^{m^*}S_N(Z^{(i)}), \tag{28}$$

where $S_N(Z^{(i)})=S_N(Z_1^{(i)},\dots,Z_N^{(i)})$ is the final value of the asset price. The expression in Eq. (28) 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 Eqs. (25) and (28), we obtain the SubSim estimate of the option price:

$$P\approx\widehat{P}^{\text{SubSim}}=\hat{p}_E^{\text{SubSim}}(\widehat{\mathbb{E}}_{\mathbb{Q}}^{\text{SubSim}}-K). \tag{29}$$

SubSim as described above, yields an estimator for the execution probability, p_E, which scales like a power of the logarithm of p_E (Au & Beck, 2001):

$$\delta\big(\hat{p}_E^{\text{SubSim}}\big)=\sqrt{\frac{(1+\gamma)(1-\beta)}{M\beta(|\ln\beta|)^r}|\ln p_E|^r}\propto|\ln p_E|^{r/2}, \tag{30}$$

where γ is a constant that depends on the correlation of the Markov chain states and $2\leq r\leq 3$. Comparing Eq. (30) against the

[8] The MMA algorithm is a component-wise modification of the original (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) algorithm, which is specifically tailored for sampling in high dimensions, where the original algorithm is known to perform poorly (see, Katafygiotis and Zuev, 2008, for the details).

[9] For an analysis around the choice of β, the interested reader is directed to Appendix B.

[10] The total number of subsets (levels) L is not a parameter set but the user by it is decided endogenously by the method, based on how rare the event in question is. A rarer event, apparently, needs more levels to be accurately estimated.

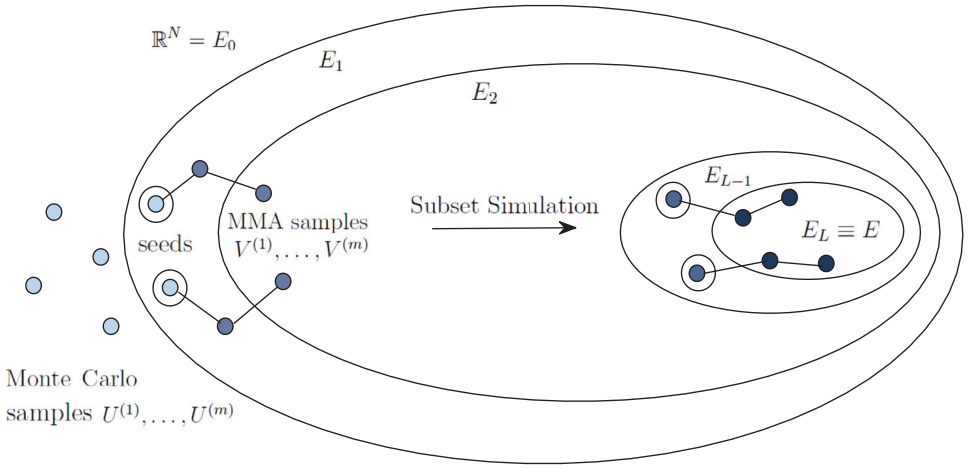

Fig. 3. Schematic illustration of Subset Simulation. First, Monte Carlo samples $U^{(1)}, \ldots, U^{(m)}$ are generated. Next, $\tilde{m} = \beta m$ "seeds" (the closest samples to E) are chosen and MMA is used to generate $V^{(1)}, \ldots, V^{(m)}$ from these seeds in the direction of E. The SubSim algorithm proceeds in this way until the target rare event E has been reached and sufficiently sampled. In this visualization, $m = 6$ and $\beta = 1/3$.

CV of a standard MCS method (Liu, 2001; Robert & Casella, 2004)

$$\delta\left(\hat{p}_E^{\mathrm{MC}}\right) = \frac{\sqrt{\mathrm{Var}\left[\hat{p}_E^{\mathrm{MC}}\right]}}{\mathbb{E}\left[\hat{p}_E^{\mathrm{MC}}\right]} = \sqrt{\frac{1-p_E}{Mp_E}} \propto p_E^{-1/2}, \tag{31}$$

reveals a serious drawback of MCS, as it makes it inefficient in estimating small probabilities of rare events. Indeed, as $p_E \to 0$, then $\delta\left(\hat{p}_E^{\mathrm{MC}}\right) \approx 1/\sqrt{Mp_E}$. 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 MCS, $\delta\left(\hat{p}_E^{\mathrm{SubSim}}\right) \ll \delta\left(\hat{p}_E^{\mathrm{MC}}\right)$. This property guaranties that SubSim produces more accurate (on average) estimates of small probabilities of rare events.

In case the asset price S 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 6 – that SubSim should be substantially more efficient in estimating prices of barrier options on high volatility assets than MC-based methods.

5. Complexity theorem

The complexity theorem provides upper bounds for the MSE and the computational complexity/cost of the SubSim estimator $\hat{P}$ for the option price P at $t = 0$, by examining their limiting behavior. Both upper bounds are given in terms of the execution probability, p_E. Note that the theorem does not make any assumptions regarding the underlying SDE or the functional of the solution used. We want to re-emphasise here that our treatment is insensitive to the choice of the underlying price dynamics.

However, before we proceed to showing the main result in this section, we introduce two lemmas in order to establish important statistical properties of the barrier option price estimator $\hat{P}$ as regards its bias and its corresponding CV.

Lemma 1. *The fractional bias of the SubSim estimator $\hat{P}$ is of order $1/M$, for every M. That is:*

$$\left|\mathbb{E}\left[\frac{\hat{P}-P}{P}\right]\right| = O\left(\frac{1}{M}\right), \tag{32}$$

where M denotes the total number of samples.

Proof. For each simulation level i, we can define the following standardized variable:

$$Z_i = \frac{\hat{P}_i - P_i}{\sigma_i} = \frac{p_i(\widehat{S_T - K}) - p_i(S_T - K)}{\sigma_i}, \tag{33}$$

where it is clear that the mean and variance equal to $\mathbb{E}[Z_i] = 0$ and $\mathbb{E}[Z_i^2] = 1$, respectively. We can solve Eq. (33) for $\hat{P}_i$ to get $\hat{P}_i = P_i + \sigma_i Z_i$. We also have that

$$\begin{aligned}\frac{\hat{P}-P}{P} &= \prod_{i=1}^{L}\frac{\hat{P}_i - P_i}{P_i} = \prod_{i=1}^{L}\frac{p_i(\widehat{S_T - K})^+ - p_i(S_T-K)^+}{p_i(S_T-K)^+} \\ &= \prod_{i=1}^{L}\frac{p_i(\widehat{S_T - K})^+}{p_i(S_T-K)^+} - 1,\end{aligned} \tag{34}$$

where L is the number of simulation levels the SubSim generates. Using Eq. (33), we can rewrite Eq. (34) as

$$\frac{\hat{P}-P}{P} = \prod_{i=1}^{L}(1+\delta_i Z_i) - 1, \tag{35}$$

where $\delta_i = \sigma_i / P_i$ denotes the CV of P_i. The RHS of Eq. (35) can be further expanded as

$$\begin{aligned}\prod_{i=1}^{L}(1+\delta_i Z_i) - 1 = &\sum_{i=1}^{L}\delta_i Z_i + \sum_{i>j}\delta_i\delta_j Z_i Z_j \\ &+ \sum_{i>j>k}\delta_i\delta_j\delta_k Z_i Z_j Z_k + \cdots + \prod_{i=1}^{L}\delta_i Z_i.\end{aligned} \tag{36}$$

The expected value of Eq. (36), using $\mathbb{E}[Z_i] = 0$, is equal to

$$\begin{aligned}\mathbb{E}\left[\frac{\hat{P}-P}{P}\right] = &\sum_{i>j}\delta_i\delta_j\mathbb{E}[Z_i Z_j] \\ &+ \sum_{i>j>k}\delta_i\delta_j\delta_k\mathbb{E}[Z_i Z_j Z_k] + \cdots + \mathbb{E}\left(\prod_{i=1}^{L}Z_i\right)\prod_{i=1}^{L}\delta_i.\end{aligned} \tag{37}$$

In case every Z is independent of each other, the expectation of Eq. (37) is equal to zero and there is no bias for $\hat{P}$. In the more general case where Z's are correlated, using that δ_i is $O(1/\sqrt{M})$ we have that the first term of the RHS of Eq. (37) is $O(1/M)$ and the rest of the terms are of order at least $O(1/(M\sqrt{M}))$. Finally, from the Cauchy inequality we have that $|\mathbb{E}[Z_i Z_j]| \leq \sqrt{\mathbb{E}[Z_i]^2\mathbb{E}[Z_j]^2}$. Ap-

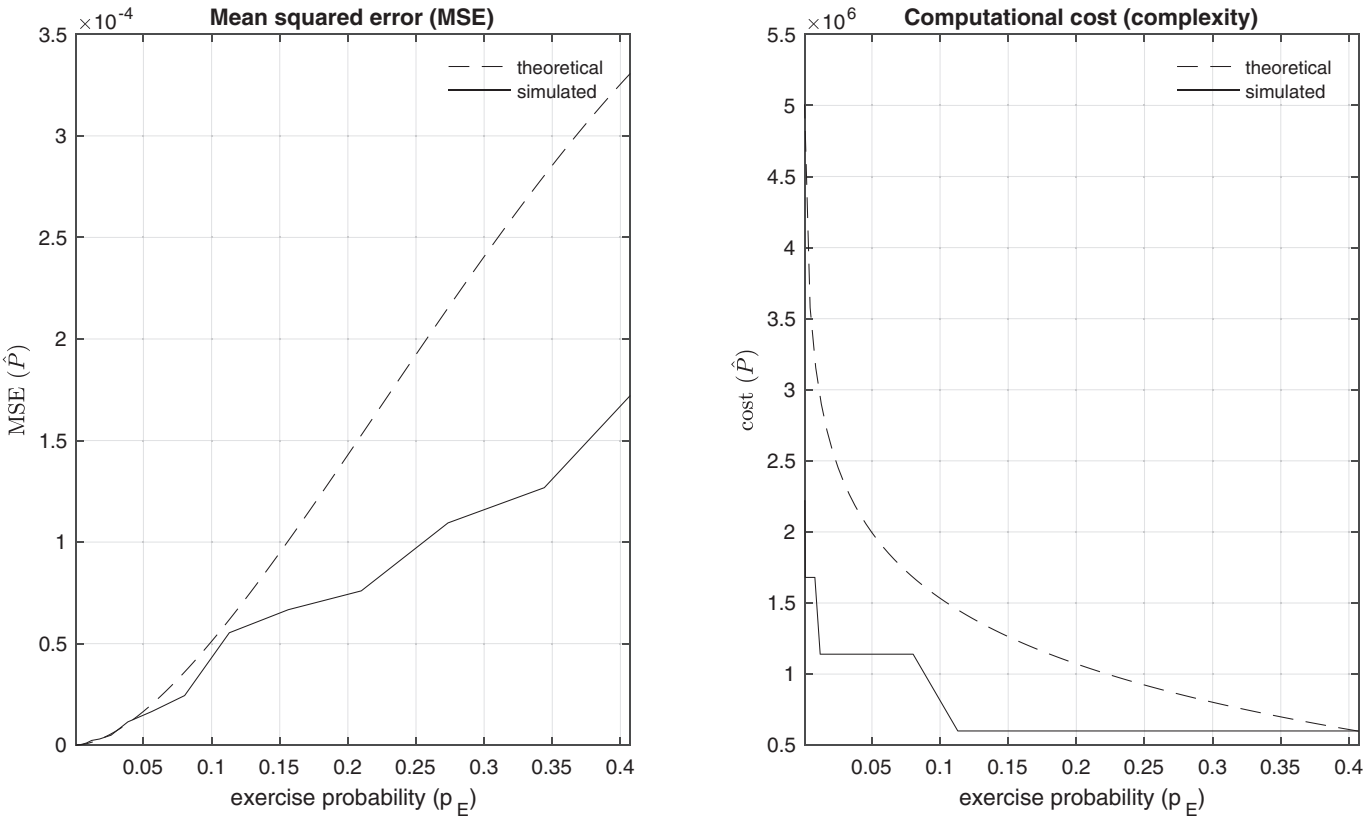

Fig. 4. Mean squared error and complexity/cost of the SubSim estimator $\hat{P}$. The simulation results on the left panel show that the MSE scales like $p_E^2|\log p_E|^{r-1}m^{-1}$, $r=2$ (in general $2\leq r\leq 3$). In accordance with the theoretical findings, simulated MSE drops with respect to p_E. Computational cost/complexity of SubSim with respect to the probability of execution is presented on the right panel. The simulation results show that the cost can be bounded above by a function of $|\log p_E|$.

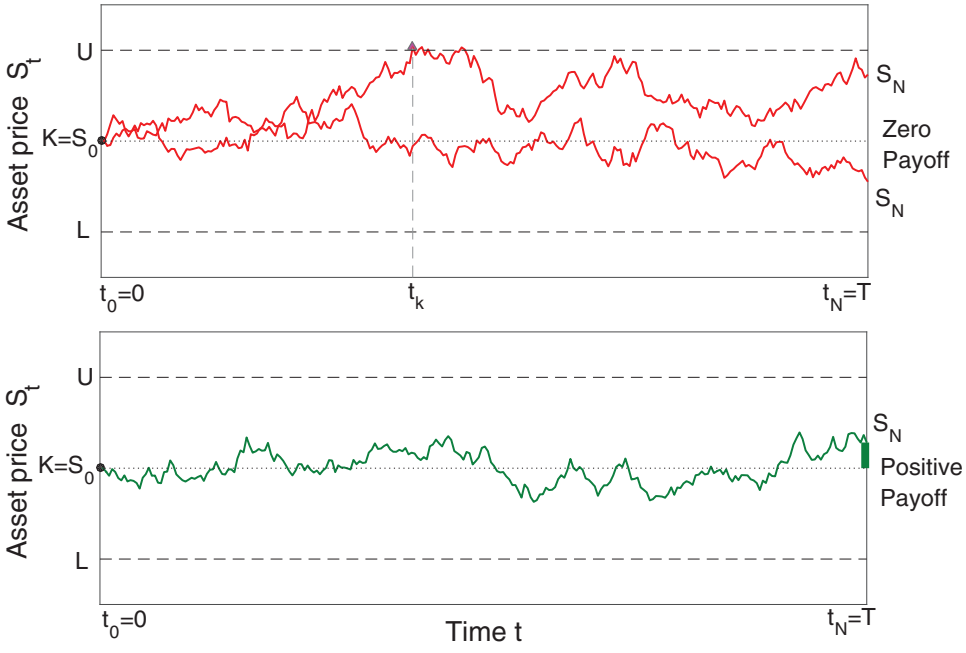

Fig. 5. Asset price trajectories. The top panel shows two asset trajectories that lead to a zero payoff: one trajectory breaks the upper barrier U at time t_k, the other ends up below the strike, $S_N < K$. The bottom panel shows an asset price trajectory that results in a positive payoff $S_N - K$. For the sake of illustration, both lower and upper barriers are constant.

Table 1
Simulation results. This table shows the mean values and coefficients of variations 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 σ. All statistics are obtained from 100 independent runs of the algorithms.

σ	$\hat{p}_E^{SubSim}/\hat{p}_E^{MCS}$	$\hat{P}_0^{SubSim}/\hat{P}_0^{MCS}$	$\delta(\hat{p}_E^{SubSim})/\delta(\hat{p}_E^{MCS})$	$\delta(\hat{P}_0^{SubSim})/\delta(\hat{P}_0^{MCS})$
0.200	8.30×10^{-3} / 8.26×10^{-3}	2.93×10^{-2} / 2.91×10^{-2}	0.030 / 0.0281	0.034 / 0.0347
0.216	4.32×10^{-3} / 4.34×10^{-3}	1.52×10^{-2} / 1.53×10^{-2}	0.032 / 0.0391	0.036 / 0.0476
0.233	2.04×10^{-3} / 2.04×10^{-3}	7.18×10^{-3} / 7.19×10^{-3}	0.039 / 0.0596	0.044 / 0.0673
0.252	8.67×10^{-4} / 8.76×10^{-4}	3.06×10^{-3} / 3.11×10^{-3}	0.048 / 0.0788	0.055 / 0.0985
0.272	3.23×10^{-4} / 3.21×10^{-4}	1.14×10^{-3} / 1.15×10^{-3}	0.057 / 0.126	0.062 / 0.160
0.294	1.06×10^{-4} / 1.08×10^{-4}	3.75×10^{-4} / 3.80×10^{-4}	0.060 / 0.217	0.069 / 0.282
0.317	2.91×10^{-5} / 2.63×10^{-5}	1.03×10^{-4} / 9.38×10^{-5}	0.076 / 0.406	0.081 / 0.476
0.343	6.85×10^{-6} / 5.66×10^{-6}	2.46×10^{-5} / 2.14×10^{-5}	0.099 / 0.759	0.109 / 1.014
0.370	1.31×10^{-6} / 9.93×10^{-7}	4.69×10^{-6} / 3.06×10^{-6}	0.153 / 1.971	0.160 / 2.337
0.400	1.99×10^{-7} / 2.45×10^{-7}	7.20×10^{-7} / 1.10×10^{-6}	0.180 / 3.844	0.205 / 4.017

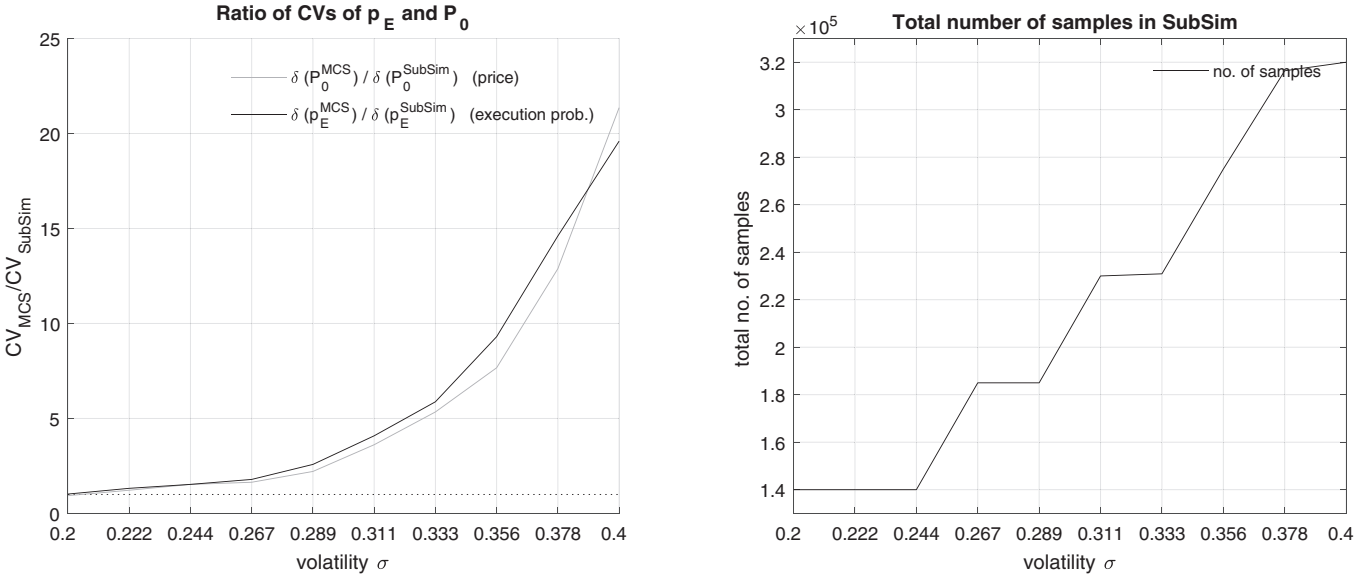

Fig. 6. Ratios of CVs. The ratios $\delta(\hat{p}_E^{MCS})/\delta(\hat{p}_E^{SubSim})$ and $\delta(\hat{P}_0^{MCS})/\delta(\hat{P}_0^{SubSim})$ versus the volatility σ are presented on the left panel. The right panel shows the total number of samples used in Subset Simulation against volatility σ when $L = 90$ and $U = 110$.

plying this result to Eq. (37) shows that

$$|\mathbb{E}\Big[\frac{\hat{P}-P}{P}\Big]| \leq \sum_{i>j}\delta_i\delta_j + o(1/M) = O\Big(\frac{1}{M}\Big). \tag{38}$$

□

Remark 1. The result in Eq. (32) should come as no surprise. Given that the payoff of the option is calculated only at the final simulation level L, the final estimation of the exercise probability, p_E, requires only one step, which adds no computational complexity to the estimation.

We next establish a result for the CV of the SubSim estimate $\hat{P}$ introducing the following lemma:

Lemma 2. *The squared CV, δ^2, of the SubSim estimator $\hat{P}$ is of order $1/M$, for every M. That is:*

$$\delta^2 = \mathbb{E}\Big[\frac{\hat{P}-P}{P}\Big]^2 = O\Big(\frac{1}{M}\Big), \tag{39}$$

where M denotes the total number of samples.

Proof. Rewriting the expectation $\mathbb{E}\big[\frac{\hat{P}-P}{P}\big]^2$ in terms of the product $\prod_{i=1}^{L}\hat{P}/P - 1 = \prod_{i=1}^{L}(1+\delta_i Z_i) - 1$, where $\delta_i = \sigma_i/P_i$, ensures that the same sequence of steps, as in the proof of Lemma 1, can be also applied here. This shows that $\hat{P}$ is a consistent estimator for P and its CV δ is $O(1/\sqrt{M})$. □

With the two lemmas above, we show that the SubSim estimate for the option price $\hat{P}$ has the same statistical properties as the exercise probability, p_E. Subsequently, results established for the second can be used for the first and vice versa. We now use the results in the two lemmas to derive the upper bounds for the MSE and the computational complexity for the SubSim estimate, $\hat{P}$.

Theorem 1. *The SubSim estimator, $\hat{P}$, for a functional of the solution $\hat{S}$ to a given SDE has*

(i) *the MSE bounded from above by $c_1 p_E^2 |\log p_E|^{r-1} m^{-1}$,*
(ii) *with computational cost which has an upper bound of $c_2(|\log p_E|)mh_s^{-1}$,*

where c_1, c_2 are constants, p_E is the probability of positive payoff at maturity, m is the number of samples per subset, $2 \leq r \leq 3$ is a parameter dependent on the correlation between the intermediate execution probabilities, and h_s is the time-step used in the discretization of the given SDE.

Proof. (i) From Eq. (30), we have that the squared CV of the exercise probability, p_E, is equal to

$$\delta^2 = \frac{(1+\gamma)(1-\beta)}{\beta|\log\beta|^r M}|\log p_E|^r, \tag{40}$$

where γ is a constant related to the correlation between the states of the Markov chains used for the sampling at different levels, β is the level probability and M is the total number of samples used by

the method. Choosing $L = \log p_E / \log \beta$ the following holds:

$$\begin{aligned}\delta^2 &= \frac{(1+\gamma)(1-\beta)}{\beta L m}|L|^r \\ &= \frac{(1+\gamma)(1-\beta)}{\beta}\left|\frac{\log p_E}{\log \beta}\right|^{r-1} m^{-1} \\ &= O(|\log p_E|^{r-1} m^{-1}),\end{aligned} \tag{41}$$

using now that the total number of samples, $M = Lm$, where L is the number of subsets (levels) and m the samples per subset. To estimate the upper bound for the MSE of $\hat{P}$, we notice that

$$\delta_{\hat{P}} = \frac{\sqrt{\mathrm{Var}[\hat{P}]}}{\mathbb{E}[\hat{P}]} = \frac{\sqrt{\mathrm{MSE}[\hat{P}] - \mathrm{Bias}[\hat{P}, P]^2}}{\mathbb{E}[\hat{P}]}. \tag{42}$$

Squaring both sides of Eq. (42) gives

$$\delta^2_{\hat{P}} = \frac{\mathrm{MSE}[\hat{P}] - \mathrm{Bias}[\hat{P}, P]^2}{\mathbb{E}[\hat{P}]^2}, \tag{43}$$

which equivalently can be written as

$$\mathrm{MSE}[\hat{P}] \approx \delta^2_{\hat{P}} \mathbb{E}[\hat{P}]^2 + \mathrm{Bias}[\hat{P}, P]^2. \tag{44}$$

To bound the MSE in Eq. (44), we can make use of Lemmas 1 and 2 which show that the first term of the MSE is $O(1/M)$ while the second term is $O(1/M^2)$. This results in a MSE bounded above by p_E^2/M as for large values of M the first term prevails. The term p_E^2 enters the upper bound of the MSE from the estimation of the option price $\hat{P} = \mathbb{E}[p_E(S_T - K)]$ at the final level L. Moreover, in Eq. (41), we show that δ^2 is $O(|\log p_E|^{r-1} m^{-1})$ which derives the following upper bound for the MSE of $\hat{P}$:

$$\mathrm{MSE} \leq c_1 \frac{p_E^2}{|\log p_E|^{1-r} m}. \tag{45}$$

(ii) To estimate the complexity (or computational cost) involved in the estimation of $\hat{P}$, we make use of the result that

$$L = \frac{|\log p_E|}{\log \beta} = O(|\log p_E|), \tag{46}$$

to approximate the cost by

$$C \approxeq L m h_s^{-1}, \tag{47}$$

where h_s is the discretization step. From Eq. (47) we can bound the computational cost above by

$$C \leq c_2 (|\log p_E|) m h_s^{-1}. \tag{48}$$

□

The result in (*i*) 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 proportional to the natural logarithm of the execution probability, p_E. As the execution probability decreases, the absolute value of its logarithm increases, resulting in a higher computational cost. Indeed, as expected, the smaller the execution probability, the more computationally demanding the

Table 2
Simulation results for the GBM. Barrier option prices, CV and CPU running times for MCS and SubSim are presented in the top panel. Results from SubSim and MLMC for the case of a single and double barrier option are presented in the bottom panel.

	Monitoring frequency											
	N = 25						$N = 125$					
	option price		CV		CPU time		option price		CV		CPU time	
σ	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim
0.10	0.8506 (0.0008)	0.8486 (0.0008)	0.0096	0.0095	7s	17s	0.6978 (0.0008)	0.6983 (0.0008)	0.0114	0.0125	22s	81s
0.15	0.3010 (0.0005)	0.3023 (0.0004)	0.0183	0.0150	10s	45s	0.1944 (0.0004)	0.1941 (0.0003)	0.0230	0.0165	20s	214s
0.20	0.0891 (0.0003)	0.0898 (0.0002)	0.0332	0.0228	6s	42s	0.0375 (0.0002)	0.0375 (0.0001)	0.0562	0.0322	21s	217s
0.25	0.0230 (1.609e−04)	0.0229 (7.564e−05)	0.0698	0.0330	7s	77s	0.0053 (7.931e−05)	0.0053 (2.174e−05)	0.1479	0.0412	20s	329s
0.30	0.0054 (7.554e−05)	0.0054 (2.328e−05)	0.1381	0.0432	6s	70s	5.363e−04 (2.316e−05)	5.628e−04 (3.9919e−6)	0.4319	0.0709	21s	450s
0.35	1.180e−03 (3.637e−05)	1.197e−03 (6.763e−6)	0.3081	0.0564	6s	96s	3.932e−05 (6.029e−6)	4.704e−05 (4.203e−7)	1.5331	0.0893	19s	573s
0.40	2.634e−04 (1.540e−05)	2.599e−04 (1.827e−6)	0.5847	0.0703	7s	124s	5.960e−6 (2.384e−6)	3.1783e−6 (4.832e−8)	4.006	0.1519	20s	827s
	Single Barrier						Double Barrier					
	option price		CV		CPU time		option price		CV		CPU time	
σ	SubSim	MLMC	SubSim	MLMC	SubSim	MLMC	SubSim	MLMC	SubSim	MLMC	SubSim	MLMC
0.10	9.1972 (0.0040)	9.0358 (0.0004)	0.0044	0.0004	80s	150s	0.6977 (0.0008)	0.7060 (0.0004)	0.0112	0.0058	56s	37s
0.15	8.6094 (0.0061)	8.2155 (0.0004)	0.00711	0.0005	81s	279s	0.1937 (0.0003)	0.2250 (0.0003)	0.0164	0.0174	140s	36s
0.20	8.2011 (0.0072)	7.720 (0.0003)	0.0088	0.0004	83s	332s	0.0378 (0.0001)	0.0748 (0.0004)	0.0329	0.0583	140s	37s
0.25	7.9791 (0.0075)	7.7022 (0.0003)	0.0095	0.0004	82s	505s	0.0052 (2.630e-05)	0.0278 (4.347e-04)	0.0497	0.1559	213s	35s
0.30	7.8412 (0.0090)	7.3905 (0.0004)	0.0115	0.0005	85s	788s	5.598e-04 (3.629e-6)	5.742e-03 (1.265e-04)	0.0648	0.2203	317s	37s
0.35	7.803 (0.0097)	7.4265 (0.0004)	0.0125	0.0005	84s	1,058s	4.685e-04 (4.559e-7)	2.971e-03 (6.831e-05)	0.0973	0.2299	420s	36s
0.40	7.7991 (0.01170)	7.5367 (0.0004)	0.0150	0.0005	85s	1,445s	3.198e-6 (4.627e-8)	9.937e-04 (2.450e-05)	0.1446	0.2466	550s	36s

estimation of $\hat{P}$ is. Fig. 5 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.

6. Simulation study

6.1. 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. For instance, our method can be very easily adjusted to accommodate single barrier options, or barrier put options, while it can even account for options with varying barriers. All simulation runs where conducted on a Intel i7 - 6700 with x64-based processor and CPU speed at 3.40 Ghz.

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 Eq. (7), which takes a non–zero value only in case the asset price trajectory remains within the two barriers and ends up above the strike price K. For illustrative purposes, Fig. 5 shows several asset trajectories that lead to both option expiration and positive payoff.

6.2. Simulation results for SubSim vs standard MCS

In the first of our numerical experiments, we consider a double knock-out barrier call option with a starting price (spot) $S_0 = 100$ that evolves under the GBM in Eq. (1), 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 $[0, T]$ at equally spaced times $0 = t_0 < t_1 < \ldots < t_N = 1$ with frequency $\Delta t = T/N$, where $N = 250$ which implies daily monitoring of the barrier option price. We further assume that the drift of the underlying asset is constant $\mu = 0.10$. To observe the effect of high volatility, we vary the value of σ over ten different values logarithmically spaced between $\sigma_{\min} = 0.20$ and $\sigma_{\max} = 0.40$.

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

$$P_0 = P \exp\left(- \int_0^T r(t) dt \right), \tag{49}$$

where P is the value of the option at the end of time period given by Eq. (7) and estimated by Eq. (29), $e^{-\int_0^T r(t)dt}$ 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, $r = 0.10$.

Table 3
Simulation results for the double exponential jump diffusion model. Barrier option prices, CV and CPU running times for the case of double exponential jump diffusion with jump rates $\lambda = 1$ (i.e., one jump expected between zero and T) and $\lambda = 3$ and two different monitoring frequencies, $n = 25$ and $n = 125$ times between zero and T. Volatility σ ranges between 0.10 and 0.40 where the benefits from using the SubSim method over Monte Carlo are more evident.

	Monitoring Frequency											
	$N = 25$						$N = 125$					
	option price		CV		CPU time		option price		CV		CPU time	
σ	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim	MC	SubSim
	jump intensity $\lambda = 1$											
0.10	0.3669 (0.0006)	0.3656 (0.0006)	0.0155	0.0162	9s	28s	0.2943 (0.0004)	0.2943 (0.0005)	0.0148	0.01812	39s	139s
0.15	(0.1324 (0.0003)	0.1323 0.0004)	0.0262	0.0293	8s	29s	0.0826 (0.0003)	0.0823 (0.0002)	0.0356	0.0350	39s	153s
0.20	0.0399 (0.0002)	0.0395 (0.0002)	0.0542	0.0510	8s	27s	0.0161 (0.0001)	0.0162 (0.0001)	0.0862	0.0686	39s	133s
0.25	0.0103 (0.0001)	0.0103 (0.0001)	0.1053	0.1023	7s	29s	0.0022 (4.99e-05)	0.0023 (3.11e-05)	0.2326	0.1290	38s	980s
0.30	0.0025 (5.7523e−05)	0.0026 (4.0405e−05)	0.2346	0.1563	8s	197s	2.185e-04 (1.531e−05)	2.629e-04 (5.1525e−6)	0.7007	0.1959	36s	985s
0.35	5.344e−04 (2.567e−05)	5.834e−04 (9.9581e−6)	0.4801	0.1678	9s	198s	2.868e−05 (4.555e−6)	2.353e−5 (2.515e−7)	2.013	0.2963	37s	976s
0.40	1.3750e−04 (1.37e−05)	1.3665e−04 (3.02e−6)	0.99672	0.22101	8s	205s	– –	6.3643e−7 (2.7925e−8)	NaN	0.43878	37s	990s
	jump intensity, $\lambda = 3$											
0.10	0.0774	0.0775	0.0349	0.03710	9s	28s	0.0530	0.05346	0.0414	0.0404	38s	130s
	(0.0003)	(0.0003)					(0.0002)	(0.0002)				
0.15	0.0287 (0.0002)	0.02904 (0.0002)	0.0554	0.0542	8s	28s	0.0150 (0.0001)	0.0151 (0.0001)	0.0912	0.0779	38s	136s
0.20	0.0087 (9.407e−05)	0.0086 (9.679e−05)	0.10691	0.11138	8s	27s	0.0021 (4.586e−05)	0.0015 (2.266e−05)	0.21829	0.1491	35s	970s
0.25	0.0023 (3.9374e−6)	0.0024 (5.027e−05)	0.2199	0.1518	8s	198s	4.110e−04 (2.089e−05)	3.505e−04 (8.4622e−6)	0.5081	0.2484	36s	974s
0.30	5.898e−04 (2.457e−05)	5.2492e−04 (1.102e−05)	0.4166	0.2141	8s	200s	7.663e−05 (8.657e−6)	5.139e−05 (3.747e−6)	1.1297	0.7291	36s	1,617s
0.35	1.360e−04 (1.152e−05)	1.371e−04 (3.034e−6)	0.8468	0.2212	8s	195s	3.545e−6 (1.885e−6)	2.083e−7 (1.446e−8)	5.3177	0.6942	36s	2,400s
0.40	2.212e−05 (5.2443e−06)	1.439e−05 (6.1812e−08)	2.3702	0.42937	8s	372s	– –	6.833e−9 (2.658e−9)	NaN	0.7677	36s	2,412s

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,

$$P_0 \approx \widehat{P}_0^{\text{SubSim}} = \widehat{P}^{\text{SubSim}} e^{-rT}. \tag{50}$$

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 σ increases, the event of having a positive payoff becomes increasingly rare (e.g. if $\sigma = 0.40$, then $p_E \approx 2 \times 10^{-7}$), and thus, the option becomes cheaper. The right plot in Fig. 6 shows the average (based on 100 runs) total number of samples M used by SubSim versus the volatility σ. The obtained trend is again expected: as σ increases, the probability p_E becomes smaller, and, therefore, the number L of subsets in Eq. (23) increases, which leads to the increase in the total number of samples Eq. (24).

Next, we use MCS to estimate p_E and P_0. To ensure fair comparison of the two methods, for each value of σ, MCS is implemented with the same total number of samples as in SubSim. The mean values of Monte Carlo estimates for the execution probability $\hat{p}_E^{\text{MCS}}$ and the option price $\widehat{P}_0^{\text{MCS}} = \widehat{P}^{\text{MCS}} e^{-rT}$, with their CVs are presented in Table 1. The mean values of $\hat{p}_E^{\text{MCS}}$ and $\widehat{P}_0^{\text{MCS}}$ are approximately the same as those of $\hat{p}_E^{\text{SubSim}}$ and $\widehat{P}_0^{\text{SubSim}}$, which confirms that SubSim estimates are approximately unbiased. The CVs, however, differ drastically. Namely, $\delta(\hat{p}_E^{\text{SubSim}})$ and $\delta(\widehat{P}_0^{\text{SubSim}})$ are substantially smaller than $\delta(\hat{p}_E^{\text{MCS}})$ and $\delta(\widehat{P}_0^{\text{MCS}})$, respectively. This effect is more pronounced the larger the volatility. For example, if $\sigma = 0.40$, 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 4, 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 Fig. 6 we plot the ratios of CVs $\delta(\hat{p}_E^{\text{MCS}})/\delta(\hat{p}_E^{\text{SubSim}})$ and $\delta(\widehat{P}_0^{\text{MCS}})/\delta(\widehat{P}_0^{\text{SubSim}})$ versus σ. 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 MCS for the estimation of the execution probability and the option price are those for which the corresponding value of $\delta(\hat{p}_E^{\text{MCS}})/\delta(\hat{p}_E^{\text{SubSim}})$ or $\delta(\widehat{P}_0^{\text{MCS}})/\delta(\widehat{P}_0^{\text{SubSim}})$ lies above the horizontal line $y = 1$ (dotted line in Fig. 6). At that level, both methods would exhibit the same level of accuracy measured by the CV, since δ^{MCS} would equal δ^{SubSim}. We notice that SubSim outperforms MCS in every examined case as both lines (for $\hat{P}_0$ and $\hat{p}_E$) lie above the $y = 1$ level.

In the second of our simulation tests, we again assume that the price S_t of the underlying asset evolves according to the GBM in Eq. (1), but we consider different monitoring frequencies for the price of the barrier option. Specifically, the upper panel of Table 2 shows numerical results for two monitoring frequencies (i.e., $N = 25$ or $N = 125$), and seven different volatility levels ranging between 0.10 and 0.40, while the CV, standard errors and CPU times for SubSim and MCS are also reported. At first, we immediately observe the impact of the monitoring frequency on the price of the barrier option; not surprisingly, as the number of monitoring points increases, the option becomes cheaper. This stems from the increased probability of capturing the asset price trajectory in the area either above the upper or below the lower barrier, which subsequently drives the survival probability p_E significantly lower. Evidently, our main finding remains valid: as volatility increases the efficiency benefit obtained by the SubSim against MCS becomes larger. This result is insensitive to whether we consider the CV or standard errors (stated in parentheses below the option price estimates in Table 2) as again, both methods seem to produce unbiased estimates.

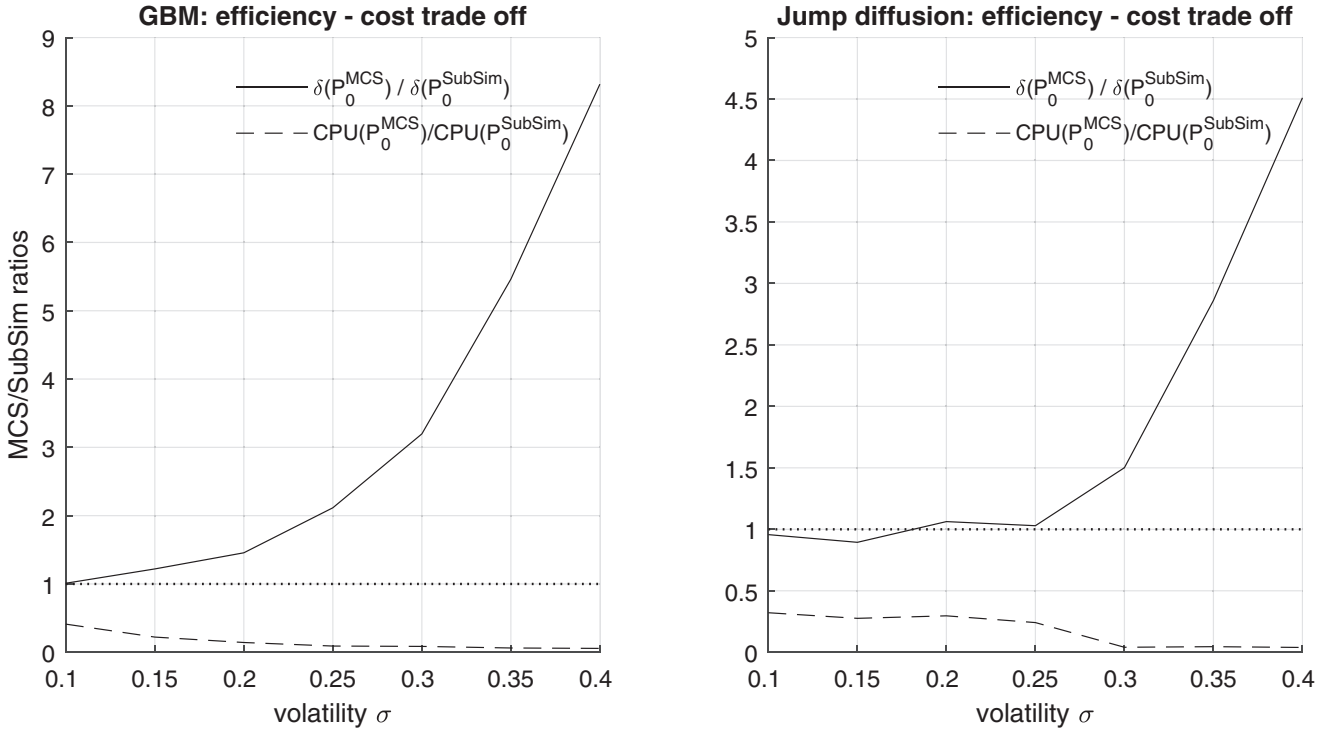

Fig. 7. Ratios of CVs and CPU times of MCS and SubSim. Ratios of the CVs δ for the option price P_0 (solid line) and the CPU times (dashed line) for the two simulation methods and the two price dynamics. The $\text{CPU}(P_0^{\text{MCS}})/\text{CPU}(P_0^{\text{SubSim}})$ ratio expresses the percentage of time required by the MCS for the same number of simulation runs, relative to that of SubSim. The trade-off between efficiency and execution time is clear: as efficiency of SubSim over MCS increases so does the CPU time of the former over the latter.

In terms of computational times (CPU times), direct MCS exhibits a relatively stable behaviour across all volatility values, while the CPU time of SubSim increases with volatility as the estimation problem converts to the rare event estimation case. In essence, MCS does not come with any provision regarding whether we are faced with a rare event estimation problem or not; a typical MCS algorithm will follow the same simulation steps for any volatility level as in fact the only factor it can vary is the number of samples it draws for every numerical experiment. On the contrary, when SubSim is confronted with a rare event estimation scenario, it attempts to sample more efficiently and moves to higher simulation levels in order to derive a much more reliable estimate for the survival probability, compared to that of MCS, simply because the method is designed for this purpose. This technicality underlies SubSim's higher computational cost compared to that of direct MCS. A potential remedy to this could be a more efficient coding of the simulation functions using a parallelized set of routines.

Focusing our attention on the case of jump diffusion, Table 3 shows that the results obtained for the GBM case hold for the former of our price dynamics equations too. Specifically, in the third of our numerical experiments, we estimate barrier option prices for two jump intensities, $\lambda = 1$ or $\lambda = 3$ and the same volatility levels as in the GBM. Not surprisingly, the barrier option under the double exponential jump diffusion in Eq. (3) is cheaper than that in GBM case. When jumps are involved in the determination of the asset price trajectory, the probability of trajectory to cross either of the two barriers increases and as a result the (already small) survival probability p_E decreases further. Nevertheless,

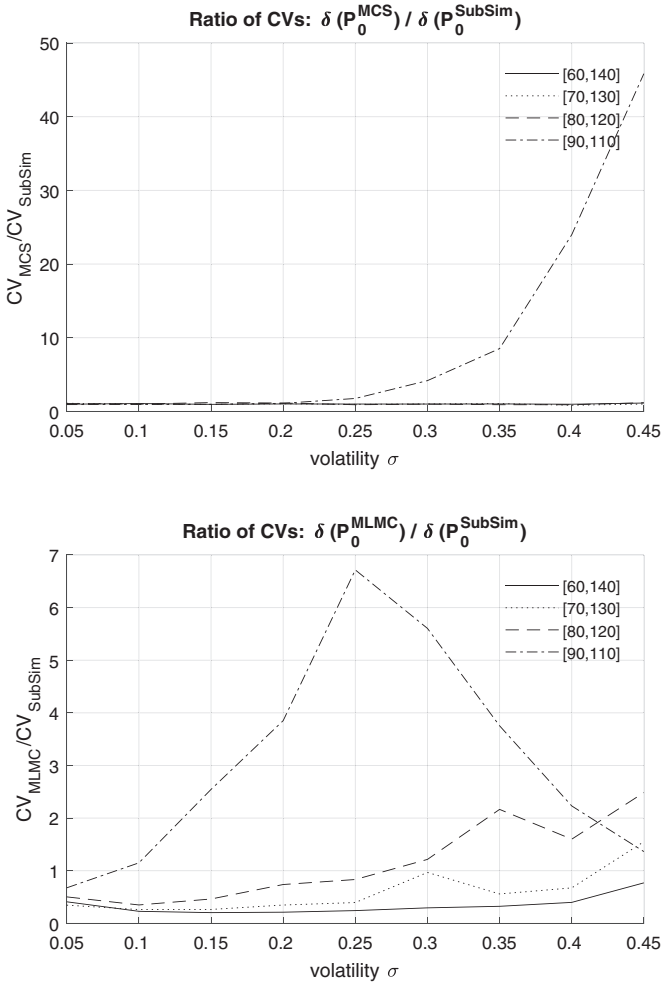

Fig. 8. Ratios of CVs of the option price P_0. The results are plotted with respect to asset volatility, for SubSim against MC (top) and SubSim against MLMC (bottom). Four different barrier levels are presented (to perform the simulations we use mainly the codes provided by Mike Giles at https://people.maths.ox.ac.uk/gilesm/mlmc/ doing the necessary adjustments in file mcqmc06.m).

Table 4
Barrier option prices. A comparison of the option prices derived by each of the three methods (MCS, MLMC and SubSim) for four barrier levels against volatility.

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

Table 5
Coefficient of variation (CV). A comparison of the CVs of the barrier option price as derived by each of the three methods (MCS, MLMC, SubSim) for four barrier levels against asset's volatility.

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

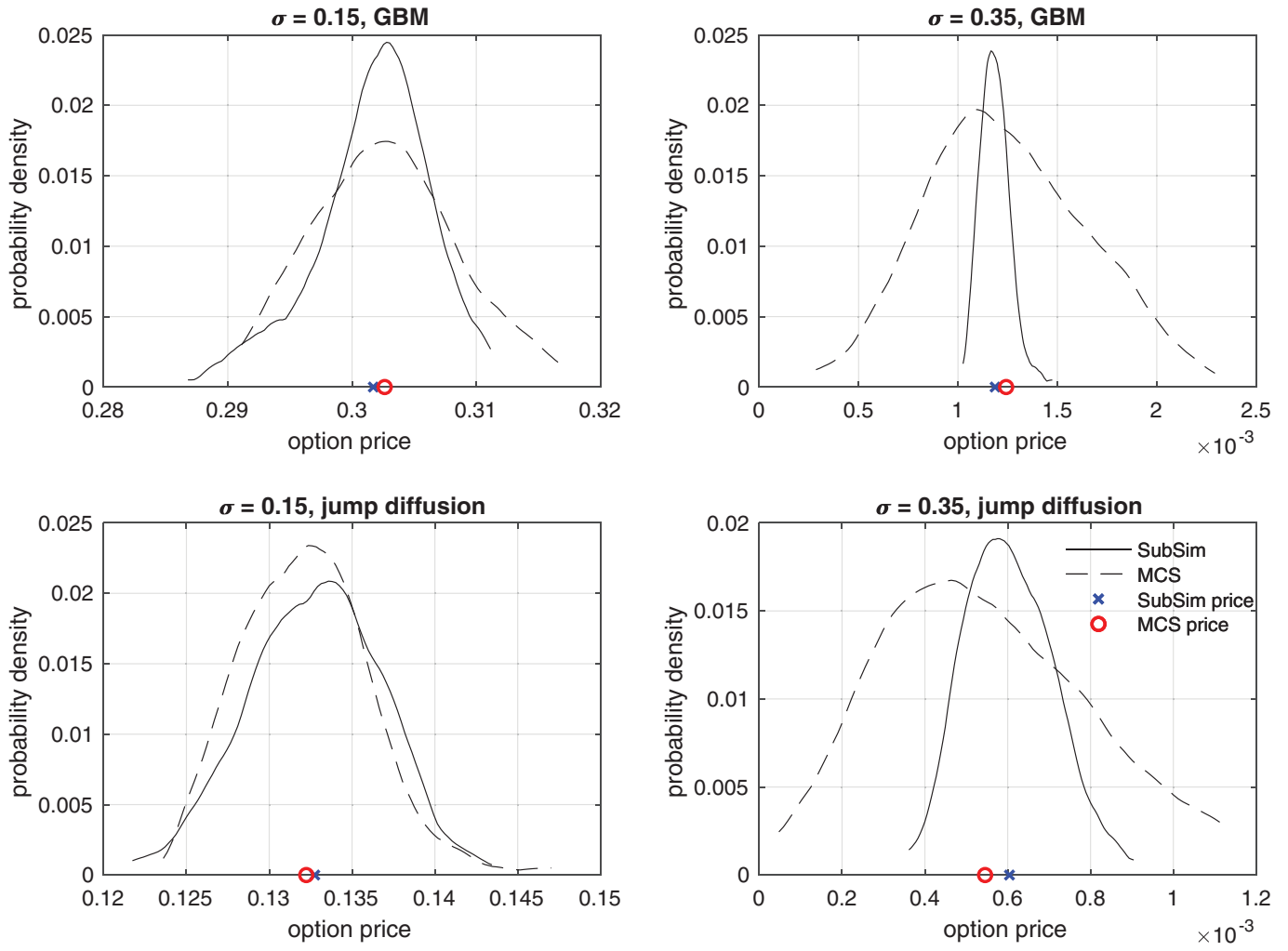

Fig. 9. Probability density of option prices. Distribution of barrier option prices for both price dynamics. The solid line shows how option prices are distributed after 100 runs of the SubSim while the dashed shows results for MCS. The benefits of using SubSim over MCS are tangible when volatility is high and consequently execution probability is low; in this case, the range of the distribution is significantly smaller and prices are concentrated around the mean option price.

this does not stop SubSim from providing us with a significantly more efficient pricing in terms of the observed CV(s) and standard errors, compared to that produced by MCS. Moreover, in line with the findings for the GBM case, the impact of the monitoring frequency on the price of the barrier option is again evident, while CPU times again follow a similar trend to that in the GBM case. In Fig. 7 we plot the trade-off between efficiency and computational cost for the two methods to show that they move to the same direction; the more efficient the pricing becomes, the higher the computational cost by SubSim as a result of its provision to deal with small survival probabilities very efficiently.

By the same token, in our final the simulation comparison tests between SubSim and MCS we increase the number of samples to $m = 200,000$ using also different levels for the lower and the upper barrier, while the assumed underlying price dynamics is the GBM. The reason we consider more samples is to compare SubSim against not only MCS but also multilevel Monte–Carlo (see Section 6.3), where $m = 200,000$ is considered in the original barrier option numerical experiments. Here, our interest is on the impact of different barrier levels on the barrier option price. Interestingly, the only barrier specification that leads to an actual rare event estimation problem is the one with barriers set at $L = 90$ and $U = 110$. In any other setup, the benefits obtained by SubSim are rather negligible, except for the case where $L = 80$, $U = 120$ and the volatility of the underlying asset takes a very large value (i.e., $\sigma \geq 0.30$) (see Fig. 8).

In general, as volatility increases, SubSim outperforms naive MCS at all barrier levels, but 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 MCS; for lower levels of σ, SubSim still outperforms MCS. The benefit of adopting SubSim is even more evident when observing Fig. 9. This plots the kernel-smoothed empirical density of the option prices after 100 runs of the MCS and the SubSim routines. Evidently, in the high-volatility scenario ($\sigma = 0.35$), the range of the distribution of the prices exported by SubSim is remarkably smaller than that of MCS, under both asset price dynamics, while prices derived by SubSim are more concentrated around the mean value.

6.3. Simulation results for SubSim vs MLMC

In this section, we compare the performance of SubSim against the MLMC simulation method of Giles (2008a,b), both for pricing single and double barrier knock-out call options under a GBM stochastic process. The parameters of this simulation exercise remain the same as in Section 6.2. The original MLMC simulation method was developed to price single knock–out barrier options, amongst other exotic derivatives, and thus we add a component

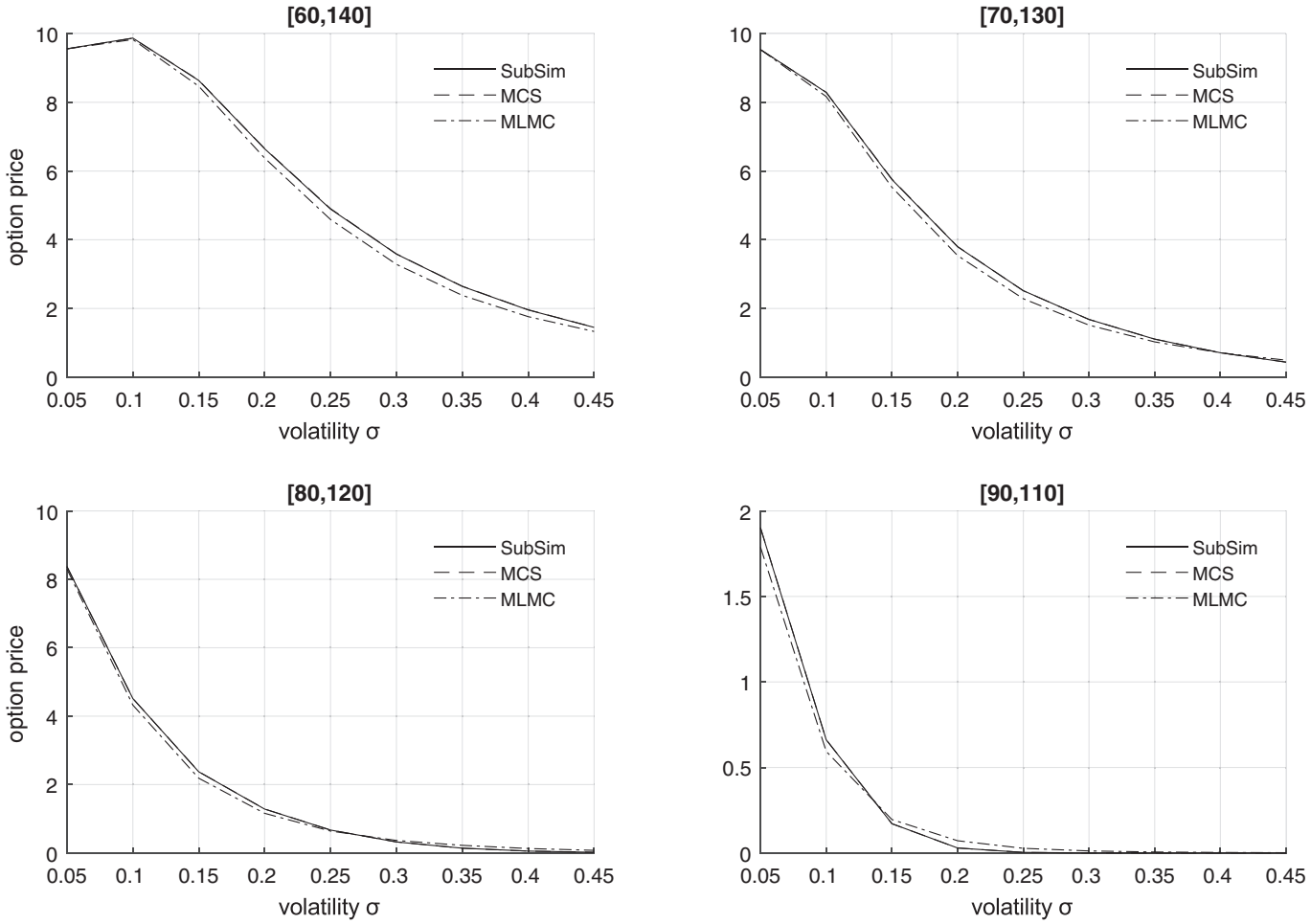

Fig. 10. Barrier option prices. Results reported for the three methods with respect to volatility. The four graphs correspond to different levels of the upper and lower barrier.

for the second barrier in order to accommodate double barrier options (see Appendices Appendix C and Appendix D). Using elements from large deviations theory (Baldi et al., 1999), we first derive the expression for the minimum of a Brownian bridge and then assuming that the event of hitting the lower or upper barrier are independent, we multiply the two corresponding execution probabilities to derive the option's execution probability at maturity.

The bottom panel of Table 2 reports results for the simulation experiments conducted to compare the performance between SubSim and MLMC. First, in the pricing of a single down-and-out barrier option with the barrier set at $B = 95$, MLMC outperforms SubSim both in terms of the resulted CV(s) and the corresponding standard errors. In essence, when we price a single barrier option, we are not faced with a rare even estimation problem, as the absence of a second barrier does not force the option to expire worthless before maturity when the underlying asset moves sharply upwards. This can be further corroborated by observing the CPU time of SubSim, which remains constant across all volatility levels. SubSim does not recognize a rare event simulation problem in any of the cases, hence it does not attempt to sample more efficiently at higher levels – which would result in larger CPU times– and subsequently its computational times are far lower than those required by MLMC for the same estimation. When it comes to pricing a double down-and-out barrier option, the estimate for the barrier option price derived by the SubSim exhibits smaller CV compared to that exported by the MLMC, especially when the underlying asset is highly volatile. MLMC appears to be a robust computational method, even when volatility increases, as its CPU time remains stable. However, SubSim's CV, although increasing with volatility, still remains lower than that of the option prices derived by the MLMC. Again, efficiency comes at a higher computational cost; in both experiments (single or double barrier) the more efficient the pricing, the larger the CPU time required by the simulation method.

In the final of the simulation experiments, we examine the impact of different barrier specifications on the barrier option prices when these are derived either by the SubSim or the MLMC. The bottom graph of Fig. 8 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 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 either to standard MCS or MLMC. The evidence we obtain here further supports the findings in Section 6.2 that SubSim is an efficient technique to price barrier options with small survival probabilities.

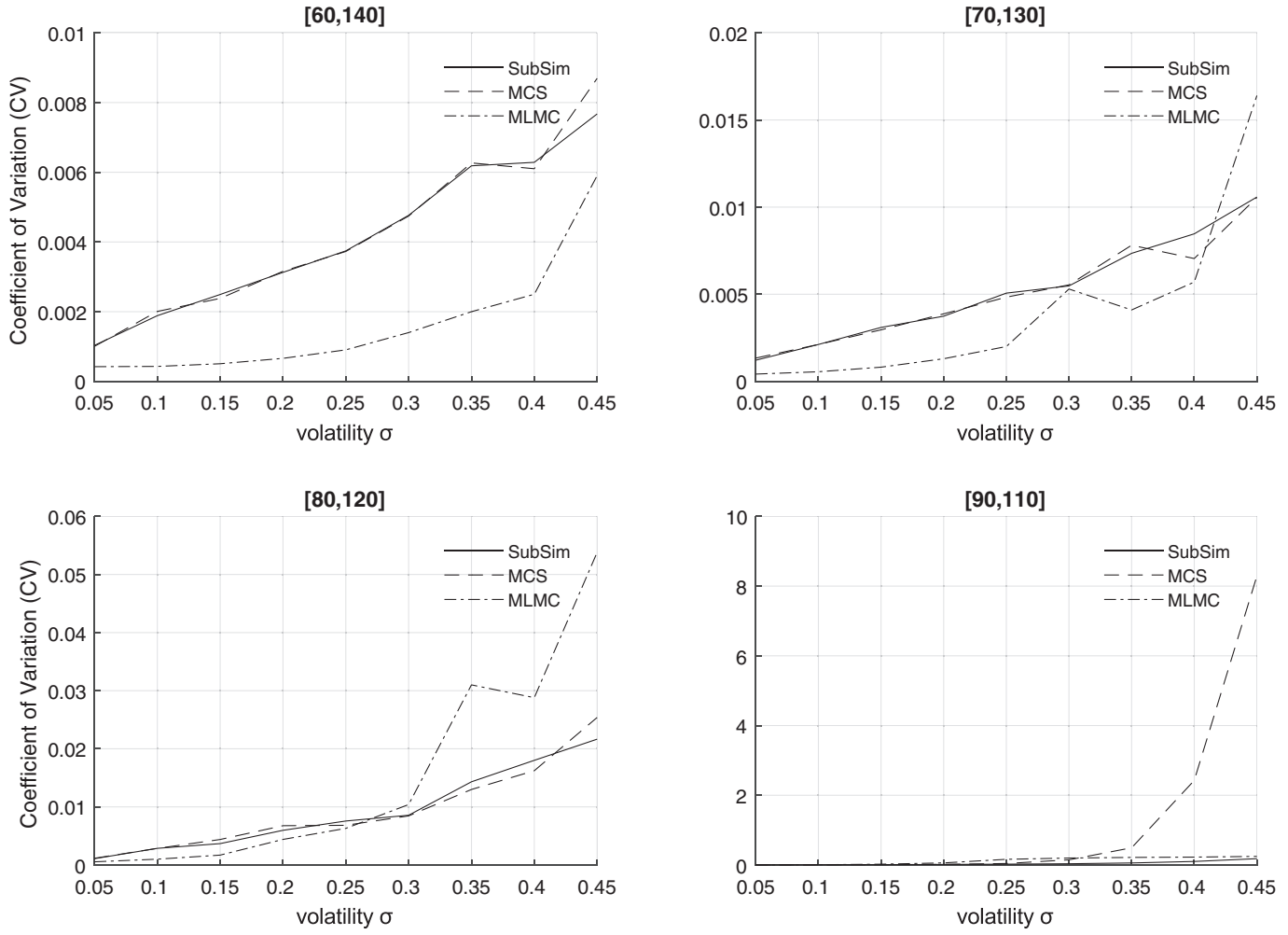

Fig. 11. Coefficient of variation (CV). Results reported for the three methods with respect to volatility for 100 runs of the pricing algorithm. The four graphs correspond to different levels of the upper and lower barrier.

Exact values for $\hat{P}_0^{\{\text{MCS,MLMC,SubSim}\}}$ (option price at $t = 0$ for each of the three methods) and $CV_{P_0}^{\{\text{MCS,MLMC,SubSim}\}}$ can be found in Tables 4 and 5, respectively. For visualization purposes, we also plot these results in Figs. 10 and 11.

7. Conclusion

In this paper, we develop a new stochastic simulation-based method for pricing barrier options. The method is based on Subset Simulation (SubSim), a very efficient algorithm for estimating small probabilities of rare events. The key observation allowing us 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 can both be estimated efficiently by SubSim. In the case of barrier options on high–volatility assets and barriers set close to the spot price of the underlying asset, SubSim is especially advantageous because of the very small probability of the contract remaining valid until maturity. We first compare the proposed SubSim method against the standard Monte Carlo simulation (MCS), under either a classical geometric Brownian motion or Kou (2002)'s double exponential jump diffusion 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 (2008a,b). Although MLMC outperforms SubSim in general, we find that SubSim can still be more efficient than MLMC in cases where the volatility of the underlying asset is high and the barriers are set close to the starting price of the asset, since this combination leads to a rare event estimation problem. As a result, the method we propose here complements MLMC, handling special cases of barrier option settings more efficiently.

In the present work, we offer a model-free framework for the pricing of double barrier options. The strengths of our methodology are its satisfactory efficiency, especially when it comes to pricing options with very small execution probability and its ability to deal with almost any underlying process. On the other side, given the focus of the method on this type of options, the computational times observed in SubSim are usually higher than those of alternative simulation-based methods.

In practice, more complicated structures and products than a single or double barrier option need to be priced and re-priced daily by practitioners in the finance industry. Options with moving barriers, basket options and option portfolios with underlying assets that evolve according to different stochastic processes or even Bermudan or American (see for example a recently proposed method in Phelan, Marazzina, & Germano, 2020) barrier options could be accommodated by the proposed method in this paper, presumably not after extensive modifications. An additional aspect

of the current study can also appear with respect to using barrier options to hedge portfolio exposures, especially in the foreign exchange and commodity markets. Given that the primary goal of a hedging exercise is to restrict the variability of a future payoff, the reduced CVs for the option price estimate that result from SubSim, incentivize deeper research around this question. Finally, an interesting extension of the present research could arise in the direction of considering an underlying process with time-varying (stochastic) volatility which is also very frequently met in practice (Cui, del Baño Rollin, & Germano, 2017), especially in an exotic option pricing problem with foreign exchange instruments used as underlying assets. The latter, as well as the study of Greeks (i.e., option price sensitivities with respect to different market factors) in the same context are currently under investigation by the authors.

Appendix A. MMA sampling from the target distribution f_z

To sample from the target distribution $f_z(z|E_{i-1})$, the MMA generates a Markov chain with stationary distribution $f_z(z|E_{i-1})$. Namely, if we let $Z^{(j)} \in E_{i-1}$ be the current state, then the next state $Z^{(j+1)}$ 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:
   $$a_k = \min\left\{1, \frac{f_k(\Psi_k)}{f_k(U_k^{(j)})}\right\}, \tag{51}$$
   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
   $$\Upsilon_k = \begin{cases} \Psi_k, & \text{with probability } a_k, \\ U_k^{(j)} & \text{with probability } 1 - a_k. \end{cases} \tag{52}$$
2. Accept or reject the candidate state:
$$U^{(j+1)} = \begin{cases} \Upsilon, & \text{if } \Upsilon \in E_{i-1}, \\ U^{(j)}, & \text{if } \Upsilon \notin E_{i-1}. \end{cases} \tag{53}$$

Appendix B. Choice of β

In our context, β represents the level probability as introduced in Au and Beck (2001), which controls for the intermediate target events. The way we choose β is not a systematic, but an adaptive one. In essence, we define β via the choice of the intermediate target thresholds α_i, in such a way that all the intermediate conditional probabilities are equal to β (i.e., to have exactly $m\beta$ samples in the target event). In practice, we choose the value of $\beta = 0.10$, which seems to work sufficiently well for most of the cases. For a sensitivity analysis on the choice of β see the table below.

Table 6 shows how the barrier option prices change with respect to the value of β for two scenarios: A low volatility one ($\sigma = 0.10$) and a high volatility one ($\sigma = 0.40$). The results are insensitive to the choice of the level probability β. As expected, increasing β results in drawing a larger number of samples from the target distribution, as we request more samples to belong to the target event. Instabilities might arise for values of β even larger than 0.20. However, such values are not reasonable choices since they result in an excessively large number of samples without tangible computational benefits. At the same time, when requiring such a large number of samples to belong to the target event, the problem becomes a degenerate rare-event estimation one, which exhibits attributes similar to that of a MCS setup.

Table 6
Variability of option prices with respect to β. Barrier option prices for difference values of the level probability β.

	$\sigma = 0.10$			$\sigma = 0.40$		
β	Price	Samples	Levels	Price	Samples	Levels
0.005	0.9672	50,000	1	0.0024	99,750	2
0.01	0.9623	50,000	1	0.0024	99,500	2
0.05	0.9659	50,000	1	0.0023	145,000	3
0.10	0.9650	50,000	1	0.0023	185,000	4
0.15	0.9640	50,000	1	0.0022	220,000	5
0.20	0.9650	50,000	1	0.0023	210,000	5

Appendix C. Probability of execution 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 execution indicator function of the barrier option in Eq. (7) can be approximated via its discrete form

$$\prod_{i=0}^{n-1} I_{\{\hat{M}_i \le U \wedge \hat{m}_i \ge L\}}, \tag{54}$$

where $\hat{M}_i$ and $\hat{m}_i$ are the maximum and minimum, respectively, of Eq. (2) in $[0, nh]$ and $T = nh$ or $h = T/n$ is the size of the timestep on a discrete grid. Eq. (54) 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 (54) 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:

$$M(t) = \max_{0 \le u \le t} S(u) \tag{55}$$

with

$$\hat{M}^h(n) = \max\{S(0), S(h), S(2h), \ldots, S(nh)\}, \tag{56}$$

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

$$m(t) = \min_{0 \le u \le t} S(u) \tag{57}$$

with

$$\hat{m}^h(n) = \min\{S(0), S(h), S(2h), \ldots, S(nh)\}, \tag{58}$$

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 $\{S(t),\ 0 \le t \le T\}$ becomes a Brownian bridge, and thus we sample from the distribution of the maximum of a Brownian bridge, a Rayleigh distribution, which results in

$$M(T) = \frac{S(T) + \sqrt{S(T)^2 - 2T\log X}}{2}, \tag{59}$$

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 Eq. (1), 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 end points S_i and S_{i+1} results in

$$M_i = \frac{S(i) + S(i+1) + \sqrt{[S(i+1) - S(i)]^2 - 2b^2 h\log X}}{2} \tag{60}$$

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

$$\hat{p}_{i,U} = P(\hat{M}_i \leq U|\hat{S}_i, \hat{S}_{i+1}) = 1 - \exp\Big(-\frac{2(U-\hat{S}_i)(U-\hat{S}_{i+1})}{b^2h}\Big), \tag{61}$$

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

$$\hat{p}_{i,U} = P(\hat{M}_i \leq U|\hat{S}_i, \hat{S}_{i+1}) = \Big\{1 - \exp\Big(-\frac{2(U-\hat{S}_i)(U-\hat{S}_{i+1/2})}{b^2h}\Big)\Big\} \times \Big\{1 - \exp\Big(-\frac{2(U-\hat{S}_{i+1/2})(U-\hat{S}_{i+1})}{b^2h}\Big)\Big\}. \tag{62}$$

Appendix D. 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

$$m_i = \frac{S(i) + S(i+1) - \sqrt{[S(i+1) - S(i)]^2 - 2b^2h\log X}}{2}, \tag{63}$$

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

$$\begin{aligned}
&P(\hat{m}_i \leq L|\hat{S}_i, \hat{S}_{i+1}) \\
&= P\Big(\frac{\hat{S}(i) + \hat{S}(i+1) - \sqrt{[\hat{S}(i+1) - \hat{S}(i)]^2 - 2b^2h\log X}}{2} \leq L|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(\sqrt{[\hat{S}(i+1) - \hat{S}(i)]^2 - 2b^2h\log X} \geq (\hat{S}(i) + \hat{S}(i+1)) - 2L|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(\hat{S}(i+1)^2 - 2\hat{S}(i)\hat{S}(i+1) + \hat{S}(i)^2 - 2b^2h\log X \\
&\geq (\hat{S}(i) + \hat{S}(i+1))^2 - 4L(\hat{S}(i) + \hat{S}(i+1)) + 4L^2|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(\hat{S}(i+1)^2 - 2\hat{S}(i)\hat{S}(i+1) + \hat{S}(i)^2 - 2b^2h\log X \\
&\geq \hat{S}(i)^2 + \hat{S}(i+1)^2 + 2\hat{S}(i)\hat{S}(i+1) - 4L(\hat{S}(i) + \hat{S}(i+1)) + 4L^2|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(-b^2h\log U \geq 2\hat{S}(i)\hat{S}(i+1) - 2L\hat{S}(i) + 2L\hat{S}(i+1) + 2L^2|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(\log U \leq -\frac{2\hat{S}_i(\hat{S}_{i+1} - L) - 2L(\hat{S}_{i+1} - L)}{b^2h}|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(\log U \leq -\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= P\Big(U \leq \exp\Big(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\Big)|\hat{S}_i, \hat{S}_{i+1}\Big) \\
&= \exp\Big(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\Big).
\end{aligned} \tag{64}$$

The probability in Eq. (64) 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

$$\hat{p}_{i,L} = 1 - \exp\Big(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\Big), \tag{65}$$

and the probability of the asset price to remain within the barriers on $[0, T]$ is equal to

$$\hat{p}_i = \hat{p}_{i,U}\hat{p}_{i,L} = \Big\{1 - \exp\Big(-\frac{2(U-\hat{S}_i)(U-\hat{S}_{i+1})}{b^2h}\Big)\Big\} \times \Big\{1 - \exp\Big(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1} - L)}{b^2h}\Big)\Big\}. \tag{66}$$

The calculation of the probability of survival for the coarse path estimation follows trivially from Eq. (66) by adjusting it using Eq. (62). 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

$$\hat{p}_i = \Big\{1 - \exp\Big(-\frac{2(U-\hat{S}_i)(U-\hat{S}_{i+1/2})}{b^2h}\Big)\Big\} \times \Big\{1 - \exp\Big(-\frac{2(U-\hat{S}_{i+1/2})(U-\hat{S}_{i+1})}{b^2h}\Big)\Big\} \tag{67}$$

$$\times \Big\{1 - \exp\Big(-\frac{2(\hat{S}_i - L)(\hat{S}_{i+1/2} - L)}{b^2h}\Big)\Big\} \times \Big\{1 - \exp\Big(-\frac{2(\hat{S}_{i+1/2} - L)(\hat{S}_{i+1} - L)}{b^2h}\Big)\Big\}. \tag{68}$$

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., & Stuart, A. M. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science, 32*(3), 405–431.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Ebens, H. (2001). The distribution of realized stock return volatility. *Journal of Financial Economics, 61*(1), 43–76.

Au, S. K., & Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics, 16*(4), 263–277.

Au, S. K., & Beck, J. L. (2003). Important sampling in high dimensions. *Structural Safety, 25*(2), 139–163.

Au, S. K., & Wang, Y. (2014). *Engineering risk assessment and design with subset simulation*. John Wiley & Sons Singapore Pte. Ltd..

Baldi, P., Caramellino, L., & Iovino, M. G. (1999). Pricing general barrier options: A numerical approach using sharp large deviations. *Mathematical Finance, 9*(4), 293–321.

Bandi, C., & Bertsimas, D. (2014). Robust option pricing. *European Journal of Operational Research, 239*(3), 842–853.

Beaglehole, D. R., Dybvig, P. H., & Zhou, G. (1997). Going to extremes: Correcting simulation bias in exotic option valuation. *Financial Analysts Journal, 53*(1), 62–68.

Bengtsson, T., Bickel, P., & Li, B. (2008). *Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems*. In D. Nolan, & T. Speed (Eds.) (2, pp. 316–334)). Institute of Mathematical Statistics.

Beskos, A., Crisan, D., & Jasra, A. (2014). On the stability of sequential monte carlo methods in high dimensions. *Annals of Applied Probability, 24*(4), 1396–1445.

Boyle, P. P., & Tian, Y. (1998). An explicit finite difference approach to the pricing of barrier options. *Applied Mathematical Finance, 5*(1), 17–43.

Boyle, P. P., & Tian, Y. (1999). Pricing lookback and barrier options under the CEV process. *Journal of Financial and Quantitative Analysis, 34*(2), 241–264.

Broadie, M., Glasserman, P., & Kou, S. (1997). A continuity correction for discrete barrier options. *Mathematical Finance, 7*(4), 325–349.

Buchen, P., & Konstandatos, O. (2009). A new approach to pricing double-barrier options with arbitrary payoffs and exponential boundaries. *Applied Mathematical Finance, 16*(6), 497–515.

Carr, P., & Chou, A.. Hedging complex barrier options. Working Paper, New York University, U.S.A., https://pdfs.semanticscholar.org/ba00/b930f3b417143bec6f2a87d3f09b6ab580fe.pdf.

Chiarella, C., Kang, B., & Meyer, G. H. (2012). The evaluation of barrier option prices under stochastic volatility. *Computers & Mathematics with Applications, 64*(6), 2034–2048.

Cui, Y., del Baño Rollin, S., & Germano, G. (2017). Full and fast calibration of the Heston stochastic volatility model. *European Journal of Operational Research, 263*(2), 625–638.

Dadachanji, Z. (2015). *FX barrier options: A comprehensive guide for Industry Quants*. Palgrave MacMillan U.K..

Davydov, D., & Linetsky, V. (2001). Pricing and hedging path-dependent options under the CEV process. *Management Science, 47*(7), 949–965.

Derman, E., & Kani, I. (1996). The ins and outs of barrier options: Part 1. *Derivatives Quarterly, 2*, 55–67.

Derman, E., & Kani, I. (1997). The ins and outs of barrier options: Part 2. *Derivatives Quarterly, 3*, 73–80.

Elliott, R. J., Siu, T. K., & Chan, L. (2014). On pricing barrier options with regime switching. *Journal of Computational and Applied Mathematics, 256*, 196–210.

Feng, L., & Linetsky, V. (2008). Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach. *Mathematical Finance, 18*(3), 337–384.
Fusai, G., Abrahams, D. I., & Sgarra, C. (2006). An exact analytical solution for discrete barrier options. *Finance and Stochastics, 10*(1), 1–26.
Fusai, G., Germano, G., & Marazzina, D. (2016). Spitzer identity, Wiener-Hopf factorization and pricing of discretely monitored exotic options. *European Journal of Operational Research, 251*(1), 124–134.
Fusai, G., & Recchioni, M. C. (2007). Analysis of quadrature methods for pricing discrete barrier options. *Journal of Economic Dynamics and Control, 31*(3), 826–860.
Geman, H., & Yor, M. (1996). Pricing and hedging double-barrier options: A probabilistic approach. *Mathematical Finance, 6*(4), 365–378.
Giles, M. B. (2008a). *Improved multilevel Monte Carlo convergence using the Milstein scheme*. Springer.
Giles, M. B. (2008b). Multilevel Monte Carlo path simulation. *Operations Research, 56*(3), 607–617.
Glasserman, P. (2013). *Monte Carlo methods in financial engineering*: 53. Springer Science & Business Media.
Glasserman, P., Heidelberger, P., Shahabuddin, P., & Zajic, T. (1999). Multilevel splitting for estimating rare event probabilities. *Operations Research, 47*(4), 585–600.
Glasserman, P., & Staum, J. (2001). Conditioning on one-step survival for barrier option simulations. *Operations Research, 49*(6), 923–937.
Golbabai, A., Ballestra, L. V., & Ahmadian, D. (2014). A highly accurate finite element method to price discrete double barrier options. *Computational Economics, 44*(2), 153–173.
Guardasoni, C., & Sanfelici, S. (2016). Fast numerical pricing of barrier options under stochastic volatility and jumps. *SIAM Journal on Applied Mathematics, 76*(1), 27–54.
Hull, J. C. (2009). *Options, futures, and other derivatives*. USA: Pearson.
Jeannin, M., & Pistorius, M. (2010). A transform approach to compute prices and Greeks of barrier options driven by a class of Lévy processes. *Quantitative Finance, 10*(6), 629–644.
Jewitt, G. (2015). *FX derivatives trader school*. John Wiley & Sons.
Katafygiotis, L. S., & Zuev, K. M. (2008). Geometric insight into the challenges of solving high-dimensional reliability problems. *Probabilistic Engineering Mechanics, 23*(2–3), 208–218.
Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management Science, 48*(8), 1086–1101.
Kou, S. G. (2007). Discrete barrier and lookback options. *Handbooks in Operations Research and Management Science, 15*, 343–373.
Kou, S. G., & Wang, H. (2004). Option pricing under a double exponential jump diffusion model. *Management Science, 50*(9), 1178–1192.
Lian, G., Zhu, S. P., Elliott, R. J., & Cui, Z. (2017). Semi-analytical valuation for discrete barrier options under time-dependent Lévy processes. *Journal of Banking & Finance, 75*, 167–183.
Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. New York: Springer Verlag.
Lo, C. F., Lee, H. C., & Hui, C. H. (2003). A simple approach for pricing barrier options with time-dependent parameters. *Quantitative Finance, 3*(2), 98–107.
Luenberger, D., & Luenberger, R. (1999). *Pricing and hedging barrier options*. Investment Practice, Stanford University, EES-OR.
Merton, R. C. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science, 4*(1), 141–183.
Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics, 21*, 1087–1092.
Phelan, C. E., Marazzina, D., Fusai, G., & Germano, G. (2018). Fluctuation identities with continuous monitoring and their application to price barrier options. *European Journal of Operational Research, 271*(1), 210–223.
Phelan, C. E., Marazzina, D., Fusai, G., & Germano, G. (2019). Hilbert transform, spectral filters and option pricing. *Annals of Operations Research, 282*(1–2), 273–298.
Phelan, C. E., Marazzina, D., & Germano, G. (2020). Pricing methods for α-quantile and perpetual early exercise options based on spitzer identities. *Quantitative Finance*, 1–20.
Rambeerich, N., & Pantelous, A. A. (2016). A high order finite element scheme for pricing options under regime switching jump diffusion processes. *Journal of Computational and Applied Mathematics, 300*, 83–96.
Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. New York: Springer Verlag.
Schoutens, W., & Symens, S. (2003). The pricing of exotic options by Monte–Carlo simulations in a Lévy market with stochastic volatility. *International Journal of Theoretical and Applied Finance, 6*(8), 839–864.
Sesana, D., Marazzina, D., & Fusai, G. (2014). Pricing exotic derivatives exploiting structure. *European Journal of Operational Research, 236*(1), 369–381.
Shevchenko, P. V., & Del Moral, P. (2017). Valuation of barrier options using sequential Monte Carlo. *Journal of Computational Finance, 20*(4), 107–135.
Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review, 136*(12), 4629–4640.
Wade, B. A., Khaliq, A. Q. M., Yousuf, M., Vigo-Aguiar, J., & Deininger, R. (2007). On smoothing of the Crank–Nicolson scheme and higher order schemes for pricing barrier options. *Journal of Computational and Applied Mathematics, 204*(1), 144–158.
Zhu, Z., & De Hoog, F. (2010). A fully coupled solution algorithm for pricing options with complex barrier structures. *Journal of Derivatives, 18*(1), 9–17.
Zuev, K. M., & Katafygiotis, L. S. (2011). Modified Metropolis-Hastings algorithm with delayed rejection. *Probabilistic Engineering Mechanics, 26*, 405–412.
Zuev, K. M., Wu, S., & Beck, J. L. (2015). General network reliability problem and its efficient solution by subset simulation. *Probabilistic Engineering Mechanics, 40*, 25–35.
Zvan, R., Vetzal, K. R., & Forsyth, P. A. (2000). PDE methods for pricing barrier options. *Journal of Economic Dynamics and Control, 24*(11–12), 1563–1590.