

Application of simplest random walk algorithms for pricing barrier options

M. Krivko* M.V. Tretyakov†

November 27, 2012

Abstract

We demonstrate effectiveness of the **first-order algorithm from [Milstein, Tretyakov. Theory Prob. Appl. 47 (2002), 53-68]** in application to **barrier option pricing**. The algorithm uses the **weak Euler approximation far from barriers and a special construction motivated by linear interpolation of the price near barriers**. It is easy to implement and is universal: it can be applied to various structures of the contracts including derivatives on multi-asset correlated underlyings and can deal with various type of barriers. **In contrast** to the **Brownian bridge techniques** currently commonly used for pricing barrier options, the **algorithm tested here does not require knowledge of trigger probabilities nor their estimates**. We illustrate this algorithm via pricing a **barrier caplet, barrier trigger swap and barrier swaption**.

AMS 2000 subject classification. 65C30, 60H30, 91B28, 91B70.

Keywords. Barrier options, exotic derivatives, weak approximation of stochastic differential equations in bounded domains, Monte Carlo technique, the Dirichlet problem for parabolic partial differential equations, interest rate derivatives.

1 Introduction

Barrier option contracts are among the most traded and oldest exotic derivatives. They accommodate investors' view about the future market behavior more closely and they are generally cheaper than the corresponding plain vanilla options. Typically, a barrier option is activated (knocked in) or deactivated (knocked out) depending on whether a vector of underlying assets or their functional has crossed a specified barrier level, which itself can be a functional of the underlying assets. **Due to its attractive features, barrier optionality has been introduced in a wide range of derivatives products.** In the context of credit

*Department of Mathematics, University of Leicester, Leicester LE1 7RH, UK; E-mail: mk211@le.ac.uk

†School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, UK; E-mail Michael.Tretyakov@nottingham.ac.uk

risk, the event of default of some reference entity can be modelled as a lower barrier on the equity of the entity. This is the key idea in structural models for pricing popular credit instruments such as credit default swaps (CDSs) and credit default obligations (CDOs). More recently, barrier optionality has been also used to model contingent convertible (CoCo) bonds which were introduced to provide financial institutions with sufficient capital in times of distress or systemic risk.

Closed form solutions for barrier option prices can be obtained only in some particular settings. For instance, they are available in the case of a single underlying asset and a constant barrier within the standard Black-Scholes setup (see, e.g. [10, 19, 4, 18, 21]). In products involving a large number of dependent assets numerical approximation for pricing and hedging barrier options is usually inevitable and this can be a challenging problem.

In this paper we assume that underlying assets are modelled via multidimensional stochastic differential equations (SDEs) and we consider European-type barrier options. The arbitrage price $u(t, x)$ of such an option solves the Dirichlet problem for a linear parabolic partial differential equation. Finding this price numerically requires efficient weak approximations of diffusions in a bounded domain.

“Ordinary” numerical methods for SDEs on a finite time interval in \mathbf{R}^d are based on a time discretization [9, 11, 16]. They ensure smallness of time increments at each step, but might not ensure smallness of space increments. In [13, 12, 15] (see also [16]) a number of weak-sense approximations for SDEs in a bounded domain were proposed, in which space increments are controlled at each step so that the constructed approximation belongs to the bounded domain. Approximations of [12] (see also [16]) are based on adaptive control of a time step of numerical integration of the SDEs. A step is chosen such that (of course, aside of reaching a required accuracy) the next state of a Markov chain approximating in the weak sense the SDEs’ solution remains in the bounded domain with probability one. This leads to a decrease of the time step when the chain is close to the boundary of the domain. The chain is stopped in a narrow zone near the boundary so that values of the solution $u(t, x)$ (i.e., the option price at time t and underlyings’ price x) in this zone can be approximated accurately by the known values of the function φ on the boundary (i.e., the value of the option at the barrier). Another type of approximations was proposed in [15] (see also [16]). In the algorithm of weak order one from [15] the step of numerical integration of the SDEs is constant for points belonging to a certain time layer $t = t_k$. Far from the boundary, a Markov chain approximating the SDEs’ solution is constructed using the weak Euler scheme (i.e., using discrete random variables for approximating the Wiener increments). When a point is close to the boundary, we make an intermediate (auxiliary) step of the random walk, which preserves the point in the time layer $t = t_k$. On this auxiliary step we “flip a coin” to decide whether to terminate the chain on the boundary or jump back in the domain and continue the random walk. The construction of this step is based on the idea of linear interpolation for the solution $u(t, x)$. The algorithm is efficient and very easy to implement. Its simpler version of order

Can jump over a barrier if we use time discrtization

1/2 in the weak sense is also presented in [15, 16]. In Section 2 we recall these two algorithms from [15, 16]. Due to our knowledge, despite simplicity of these weak schemes, they have not been used in financial applications. In this paper we try to fill this gap and illustrate their applicability to pricing barrier options.

Currently, the popular numerical approach for pricing barrier options exploits the Brownian bridge technique [1, 2, 7, 20, 3] (also see [8] for a review and the references therein). It is based on simulation of a one-dimensional Brownian bridge extremum between time steps and computing analytically the associated probability of exiting the spatial domain for each time interval of the partition. It was proved in [7] that this approach realized along with the Euler scheme (which uses Gaussian random variables for simulating Wiener increments) results in an approximation of weak order one. The Brownian bridge technique relies on analytical formulas for trigger probabilities and can run into difficulties in the case of multiple barriers and/or correlated structure of the underlyings when there are no closed formulas for the distribution of extremum. Though some extensions to these more general and not uncommon problems have been considered, e.g. in [8, 20].

In contrast the simplest random walk algorithm of [15] displays a high degree of flexibility and can be applied to various structures of the contracts including derivatives on multi-asset correlated underlyings and can deal with various type of barriers, e.g. single, double and time dependent barriers. In comparison with the Brownian bridge techniques the method of [15] does not require the knowledge of the trigger probabilities nor their estimates.

In Sections 4-6 we present three examples on how to apply the algorithm from [15] for valuation of barrier options. These contracts cover the most common types of barrier options. In the first example (Section 4), we consider an algorithm for barrier derivatives where the payoff depends on a single underlying. As an illustration, we deal with pricing a barrier caplet and provide ready-for-implementation procedure which can be easily applied to similar other contracts. The second example (Section 5) is devoted to multi-asset options with barriers imposed on all or some of the correlated underlying assets. We illustrate this case by pricing a trigger swap. The last example (Section 6) is barrier contracts written on an asset that can be expressed through some other multi-asset underlying. As a specific case, we consider valuation of a barrier swaption under the LIBOR market model (LMM).

2 Simplest random walks for stopped diffusions

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, \mathcal{F}_t , $0 \leq t \leq T$, be a filtration satisfying the usual hypotheses, (w_t, \mathcal{F}_t) be an r -dimensional standard Wiener process. Let G be a bounded domain in \mathbf{R}^d and $Q = [t_0, T) \times G$ be a cylinder in \mathbf{R}^{d+1} , $\Gamma = \bar{Q} \setminus Q$ be the part of the cylinder's boundary consisting of the upper base and lateral surface. Price of barrier options with underlying modelled by a diffusion process can usually be expressed as

$$u(t, x) = E[\varphi(\tau, X_{t,x}(\tau))Y_{t,x,1}(\tau) + Z_{t,x,1,0}(\tau)] , \quad (2.1)$$

upper base = T
lateral space = spatial
boundaries of G (barriers)

where $X_{t,x}(s)$, $Y_{t,x,y}(s)$, $Z_{t,x,y,z}(s)$, $s \geq t$, is the solution of the Cauchy problem for the system of SDEs:

$$dX = (b(s, X) - \sigma(s, X)\mu(s, X)) ds + \sigma(s, X) dw(s), \quad X(t) = x, \quad (2.2)$$

$$dY = c(s, X)Y ds + \mu^\top(s, X)Y dw(s), \quad Y(t) = y, \quad (2.3)$$

$$dZ = g(s, X)Y ds + F^\top(s, X)Y dw(s), \quad Z(t) = z, \quad (2.4)$$

$(t, x) \in Q$, and $\tau = \tau_{t,x}$ is the first exit time of the trajectory $(s, X_{t,x}(s))$ to the boundary Γ . In (2.2)-(2.4), $b(s, x)$ is a d -dimensional column-vector, the $\sigma(s, x)$ is a $d \times r$ matrix, $\mu(s, x)$ and $F(s, x)$ are r -dimensional vectors, and $Y(s)$, $Z(s)$, $c(s, X)$ and $g(s, X)$ are scalars. We assume that all the coefficients in (2.2)-(2.4), the function $\varphi(t, x)$ defined on Γ and the boundary ∂G of the space domain G satisfy some regularity conditions.

We note that the value of the expectation $u(t, x)$ in (2.1) does not depend on a choice of functions $\mu(s, x)$ and $F(s, x)$. This flexibility can be used for reducing variance of the random variable $\varphi(\tau, X_{t,x}(\tau))Y_{t,x,1}(\tau) + Z_{t,x,1,0}(\tau)$ with the aim of reducing the statistical error in computing $u(t, x)$ via the Monte Carlo technique [11, 16]. For instance, if μ and F are such that

$$\sum_{i=1}^d \sigma^{ij} \frac{\partial u}{\partial x^i} + u\mu^j + F^j = 0, \quad j = 1, \dots, r, \quad (2.5)$$

then $\text{Var}[\varphi(\tau, X_{t,x}(\tau))Y_{t,x,1}(\tau) + Z_{t,x,1,0}(\tau)] = 0$ and $\varphi(\tau, X_{t,x}(\tau))Y_{t,x,1}(\tau) + Z_{t,x,1,0}(\tau) \equiv u(t, x)$ [14, 16]. As we see from (2.5), optimal μ and F require knowledge of the the solution $u(t, x)$ (i.e., the option price) to the considered problem and its derivatives (i.e., deltas) which is impractical. However, instead of the exact $u(t, x)$ in (2.5), one can use its approximation (e.g. price and deltas for a related option for which the closed-form solution is known) to find some suboptimal μ and F which can lead to variance reduction [6, 17].

To simulate (2.1)-(2.4), we need an approximation of the trajectory $(s, X(s))$ which satisfies some restrictions related to its nonexit from the domain \bar{Q} . Let us recall two algorithms for (2.1)-(2.4) from [15, 16].

We apply the weak explicit Euler approximation with the simplest simulation of noise to the system (2.2)-(2.4):

$$X_{t,x}(t+h) \approx X = x + h(b(t, x) - \sigma(t, x)\mu(t, x)) + h^{1/2}\sigma(t, x)\xi, \quad (2.6)$$

$$Y_{t,x,y}(t+h) \approx Y = y + hc(t, x)y + h^{1/2}\mu^\top(t, x)y\xi, \quad (2.7)$$

$$Z_{t,x,y,z}(t+h) \approx Z = z + hg(t, x)y + h^{1/2}F^\top(t, x)y\xi, \quad (2.8)$$

where $h > 0$ is a time-discretization step (a sufficiently small number), $\xi = (\xi^1, \dots, \xi^r)^\top$, ξ^i , $i = 1, \dots, r$, are mutually independent random variables taking the values ± 1 with probability $1/2$. Clearly, the random vector X takes 2^r different values.

Introduce the set of points close to the boundary (a boundary zone) $S_{t,h} \subset \bar{G}$ on the layer t : we say that $x \in S_{t,h}$ if at least one of the 2^r values of the vector

X is outside \bar{G} . It is not difficult to see that due to compactness of \bar{Q} there is a constant $\lambda > 0$ such that if the distance from $x \in G$ to the boundary ∂G is equal to or greater than $\lambda\sqrt{h}$ then x is outside the boundary zone and, therefore, for such x all the realizations of the random variable X belong to \bar{G} .

Since we should impose restrictions on an approximation of the system (2.2) so that it does exit from the domain \bar{G} , the formulas (2.6)-(2.8) can be used only for the points $x \in \bar{G} \setminus S_{t,h}$ on the layer t , and a special construction is required for points from the boundary zone. Let $x \in S_{t,h}$. Denote by $x^\pi \in \partial G$ the projection of the point x on the boundary of the domain G (the projection is unique assuming that h is sufficiently small and ∂G is smooth) and by $n(x^\pi)$ the unit vector of internal normal to ∂G at x^π . Introduce the random vector $X_{x,h}^\pi$ taking two values x^π and $x + h^{1/2}\lambda n(x^\pi)$ with probabilities $p = p_{x,h}$ and $q = q_{x,h} = 1 - p_{x,h}$, respectively, where

$$p_{x,h} = \frac{h^{1/2}\lambda}{|x + h^{1/2}\lambda n(x^\pi) - x^\pi|}. \quad (2.9)$$

This construction is motivated by the following observation [15]. If $v(x)$ is a twice continuously differentiable function with the domain of definition \bar{G} , then an approximation of $v(x)$ by the expectation $Ev(X_{x,h}^\pi)$ corresponds to linear interpolation and

$$v(x) = Ev(X_{x,h}^\pi) + O(h) = pv(x^\pi) + qv(x + h^{1/2}\lambda n(x^\pi)) + O(h). \quad (2.10)$$

We emphasize that the second value $x + h^{1/2}\lambda n(x^\pi)$ does not belong to the boundary zone. We also note that p is always greater than $1/2$ (since the distance from x to ∂G is less than $h^{1/2}\lambda$) and that if $x \in \partial G$ then $p = 1$ (since in this case $x^\pi = x$).

Let a point $(t_0, x_0) \in Q$. We would like to find the value $u(t_0, x_0)$. Introduce a discretization of the interval $[t_0, T]$, for definiteness the equidistant one:

$$t_0 < t_1 < \dots < t_M = T, \quad h := (T - t_0)/M.$$

To approximate the solution of the system (2.2), we construct a Markov chain (t_k, X_k) which stops when it reaches the boundary Γ at a random step $\varkappa \leq M$. The resulting algorithm can be formulated as Algorithm 2.1 given below.

It is proved in [15] (see also [16]) that under appropriate regularity assumptions on the coefficients of (2.2)-(2.4), the boundary condition $\varphi(t, x)$ in (2.1) and on the boundary ∂G Algorithm 2.1 converges with weak order one.

The next algorithm is obtained by a simplification of Algorithm 2.1: as soon as X_k gets into the boundary domain $S_{t_k,h}$, the random walk terminates, i.e., $\varkappa = k$, and $\bar{X}_\varkappa = X_k^\pi$, $Y_\varkappa = Y_k$, $Z_\varkappa = Z_k$ is taken as the final state of the Markov chain. The resulting algorithm takes the form of Algorithm 2.2.

It is proved in [15, 16] that under appropriate regularity assumptions on the coefficients of (2.2)-(2.4), the boundary condition $\varphi(t, x)$ in (2.1) and on the boundary ∂G Algorithm 2.2 converges with weak order $1/2$. We note that in one-dimension (i.e., in the case of a single underlying) Algorithm 2.2 is analogous to pricing barrier options by binary trees (see, e.g. [5]).

Algorithm 2.1 Algorithm of weak order one for (2.1)-(2.4)

- STEP 0. $X'_0 = x_0, Y_0 = 1, Z_0 = 0, k = 0.$
- STEP 1. If $X'_k \notin S_{t_k, h}$, then $X_k = X'_k$ and go to STEP 3.
If $X'_k \in S_{t_k, h}$, then either $X_k = X'^\pi_k$ with probability $p_{X'_k, h}$ or $X_k = X'_k + h^{1/2} \lambda n(X'^\pi_k)$ with probability $q_{X'_k, h}$.
- STEP 2. If $X_k = X'^\pi_k$, then STOP and $\varkappa = k$,
 $X_\varkappa = X'^\pi_k, Y_\varkappa = Y_k, Z_\varkappa = Z_k$.
- STEP 3. Simulate ξ_k and find $X'_{k+1}, Y_{k+1}, Z_{k+1}$ according to
(2.6)-(2.8) for $t = t_k, x = X_k, y = Y_k, z = Z_k$,
 $\xi = \xi_k$.
- STEP 4. If $k + 1 = M$, STOP and $\varkappa = M, X_\varkappa = X'_M, Y_\varkappa = Y_M$,
 $Z_\varkappa = Z_M$, otherwise $k = k + 1$ and return to STEP 1.
-

Algorithm 2.2 Algorithm of weak order 1/2 for (2.1)-(2.4)

- STEP 0. $X_0 = x_0, Y_0 = 1, Z_0 = 0, k = 0.$
- STEP 1. If $X_k \notin S_{t_k, h}$, then go to STEP 2.
If $X_k \in S_{t_k, h}$, then STOP and $\varkappa = k, \bar{X}_\varkappa = X^\pi_k$,
 $Y_\varkappa = Y_k, Z_\varkappa = Z_k$.
- STEP 2. Simulate ξ_k and find $X_{k+1}, Y_{k+1}, Z_{k+1}$ according to
(2.6)-(2.8) for $t = t_k, x = X_k, y = Y_k$,
 $z = Z_k, \xi = \xi_k$.
- STEP 3. If $k + 1 = M$, STOP and $\varkappa = M, \bar{X}_\varkappa = X_M, Y_\varkappa = Y_M$,
 $Z_\varkappa = Z_M$, otherwise $k = k + 1$ and return to STEP 1.
-

3 LIBOR Market Model

We will now assume that there exists an arbitrage-free market with continuous and frictionless trading taking place inside a finite time horizon $[t_0, t^*]$.

Among the most important benchmark interest rates is the London Interbank Offered Rate (LIBOR). It is based on simple (or simply compounded) interest. The forward LIBOR rate $L(t, T, T + \delta)$ is the rate set at time t for the interval $[T, T + \delta]$, $t \leq T$. If we enter into a contract at time t to borrow one unit at time T and repay it with interest at time $T + \delta$, the interest due will be $\delta L(t, T, T + \delta)$.

A simple replication argument (see, e.g., [4]) relates LIBOR rates and bond prices via the following identity

$$L(t, T, T + \delta) = \frac{1}{\delta} \left(\frac{P(t, T)}{P(t, T + \delta)} - 1 \right), \quad (3.1)$$

where $P(t, T)$ is the price at time $t \leq T$ of a default-free zero coupon bond.

For simplicity, we fix an equidistant finite set of maturities or tenor dates

$$T_0 < \dots < T_N = T^*, \quad T_i = i\delta, \quad i = 0, \dots, N, \quad (3.2)$$

where

$$\delta = (T^* - T_0)/N,$$

denotes the fixed length of the interval between tenor dates.

Let us introduce a simplified notation for the time t forward LIBOR rate for the accrual period $[T_i, T_{i+1}]$ and the payment at T_{i+1} :

$$\begin{aligned} L^i(t) &: = L(t, T_i, T_{i+1}), \\ t_0 &\leq t \leq t^* \wedge T_i, \quad t_0 < T_i \leq T^*, \quad i = 0, \dots, N-1. \end{aligned}$$

In the case of LIBOR Market Model (LMM) the arbitrage-free dynamics of $L^i(t)$ under the forward measure $\mathbb{Q}^{T_{k+1}}$ associated with the numeraire $P(t, T_{k+1})$ can be written as the following system of SDEs (see, e.g. [4, 18, 21]):

$$\frac{dL^i(t)}{L^i(t)} = \begin{cases} \sigma_i(t) \sum_{j=k+1}^i \frac{\delta L^j(t)}{1 + \delta L^j(t)} \rho_{i,j} \sigma_j(t) dt + \sigma_i(t) dW_i^{T_{k+1}}(t), & i > k, \quad t \leq T_k, \\ \sigma_i(t) dW_i^{T_{k+1}}(t), & i = k, \quad t \leq T_i, \\ -\sigma_i(t) \sum_{j=i+1}^k \frac{\delta L^j(t)}{1 + \delta L^j(t)} \rho_{i,j} \sigma_j(t) dt + \sigma_i(t) dW_i^{T_{k+1}}(t), & i < k, \quad t \leq T_i, \end{cases} \quad (3.3)$$

where $W^{T_{k+1}} = (W_0^{T_{k+1}}, \dots, W_{N-1}^{T_{k+1}})^\top$ is an N -dimensional correlated Wiener process defined on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t_0 \leq t \leq t^*}, \mathbb{Q}^{T_{k+1}})$; the instantaneous correlation structure is defined as

$$E \left[W_i^{T_{k+1}}(t) W_j^{T_{k+1}}(t) \right] = \rho_{i,j}, \quad i, j = 0, \dots, N-1; \quad (3.4)$$

and $\sigma_i(t)$, $i = 0, \dots, N-1$, are instantaneous volatilities which we assume here to be deterministic bounded functions.

Let ρ be the instantaneous correlation matrix with elements $\rho_{i,j}$. To simulate the correlated Wiener processes, we will use the pseudo-root of the correlation matrix ρ defined via the equation

$$\rho = U U^\top, \quad (3.5)$$

where U is an upper triangular matrix. Using U and introducing N -dimensional standard Wiener process, one can re-write (3.3) in the form of (2.2).

In what follows we will assume that the current time t_0 is set to 0. For convenience, we also assume a unit notional value of all the contracts we introduce below. In our numerical experiments in the next sections we take the correlation function of the form:

$$\rho_{i,j} = \exp(-\beta |T_i - T_j|). \quad (3.6)$$

4 Barrier cap/floor

In this section we consider Monte Carlo evaluation of barrier options written on a single underlying. We use a knock-out caplet for illustration, though our treatment is rather general and can be used to value different barrier option, for instance European and Parisian barrier options and options with different barriers including single, double and time-dependent barriers, both for fixed-income and equity markets.

An Interest Rate Cap is a security that allows its holder to benefit from low floating rates and be protected from high ones. Similarly, an Interest Rate Floor is an instrument designed to protect from low floating interest rates yet allow the holder to benefit from high rates. Formally, a cap price is obtained by summing up the prices of the underlying caplets, call options on successive LIBOR rates. Also, a floor is a strip of floorlets, put options on successive LIBOR rates.

A knock-out caplet pays the same payoff as a regular caplet as long as a prescribed barrier rate H is not reached from below by the corresponding LIBOR rate before the option expires. More specifically, the price at time $t \leq T_0$ of knock-out caplet set at time T_{i-1} with payment date at T_i , $i \geq 1$, with strike K and unit cap nominal value is given by

$$V_{caplet}(t) = \delta P(t, T_{i+1}) E^{\mathbb{Q}^{T_{i+1}}} \left[(L^i(T_i) - K)_+ \chi(\theta > T_i) \middle| \mathcal{F}_t \right], \quad (4.1)$$

where θ is the first exit time of $L^i(s)$, $s \geq t$, from the interval $G = (0, H)$. Let τ be the first exit time of the space-time diffusion $(s, L^i(s))$ from the domain $Q = [t, T_i] \times (0, H)$. Obviously, $\tau = \theta \wedge T_i$.

The dynamics of $L^i(s)$ under $\mathbb{Q}^{T_{i+1}}$ is (see (3.3)):

$$\frac{dL^i(s)}{L^i(s)} = \sigma_i(s) dW_i^{T_{i+1}}(s), \quad s \leq T_i. \quad (4.2)$$

Note that the correlation structure of (3.3) does not influence the price of the knock-out caplet since it does not depend on the joint dynamics of forward rates.

One can observe that the dynamics (4.2) coincides with the model of a stock price process under the risk-neutral measure in the case of zero interest rate. This means that by dropping the factor $\delta P(t, T_{i+1})$ in (4.1), the valuation of European barrier options on equity with zero interest rate and of barrier caplets under the LMM coincide.

In the considered case the price of the barrier caplet has the well-known closed-form solution:

$$\begin{aligned} V_{caplet}(t) &= V_{caplet}(t, L^i(t)) \\ &= \delta P(t, T_{i+1}) \left\{ L^i(t) [\Phi(\delta_+(L^i(t)/K, v_i)) - \Phi(\delta_+(L^i(t)/H, v_i))] \right. \\ &\quad - K [\Phi(\delta_-(L^i(t)/K, v_i)) - \Phi(\delta_-(L^i(t)/H, v_i))] \\ &\quad - H [\Phi(\delta_+(H^2/(KL^i(t)), v_i)) - \Phi(\delta_+(H/L^i(t), v_i))] \\ &\quad \left. + KL^i(t) [\Phi(\delta_-(H^2/(KL^i(t)), v_i)) - \Phi(\delta_-(H/L^i(t), v_i))] / H \right\}, \end{aligned} \quad (4.3)$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function and

$$\delta_+(x, v) = (\ln x + v^2/2)/v, \quad \delta_-(x, v) = (\ln x - v^2/2)/v, \quad (4.4)$$

and

$$v_i^2 = \int_t^{T_i} (\sigma_i(s))^2 ds.$$

This analytical result will be used in our numerical experiments to assess the performance of proposed algorithms. We note that the algorithm presented in this example can be easily extended to a more general model of underlying when the closed-form solution might be not available. In particular, there is no difficulty in including a drift term in the underlying dynamics (see also Sections 5 and 6). In the experiments we simulate

$$\tilde{V}_{caplet}(t) = V_{caplet}(t)/\delta P(t, T_{i+1}), \quad (4.5)$$

i.e. we drop $\delta P(t, T_{i+1})$ from (4.1), which does not imply any loss of generality since the caplet price can easily be recovered by multiplying $\tilde{V}_{caplet}(t)$ by the factor $\delta P(t, T_{i+1})$ observable at time t .

4.1 Algorithm

To preserve positivity of the LIBOR rate, we simulate the log dynamics corresponding to (4.2) rather than the LIBOR rate $L^i(t)$ itself. To illustrate the variance reduction technique discussed in Section 2, we complement (4.2) with the equation (cf. (2.4)):

$$dZ = F(s, L^i) dW_i^{T_{i+1}}(s), \quad Z(0) = 0, \quad (4.6)$$

with (see (2.5) and (4.5))

$$F(s, L^i) = -\sigma_i(s) \frac{\partial}{\partial L^i} \tilde{V}_{caplet}(s, L^i(s)). \quad (4.7)$$

We choose a time step $h > 0$ so that $M = T_i/h$ is an integer. We set $\ln L_0^i = L^i(0)$ and $Z_0 = 0$. The weak Euler scheme (2.6), (2.8) applied to (4.2) in the log form and (4.6) takes the form:

$$\ln L_{k+1}^i = \ln L_k^i - \frac{1}{2} (\sigma_i(t_k))^2 h + \sigma_i(t_k) \sqrt{h} \xi_{k+1}, \quad (4.8)$$

$$Z_{k+1} = Z_k + F(s, L_k^i) \sqrt{h} \xi_{k+1}, \quad (4.9)$$

where ξ_k are independent random variables distributed by the law $P(\xi = \pm 1) = 1/2$.

The boundary zone $S_{t,h}$ required for Algorithms 2.1 and 2.2 is chosen here as

$$S_{t_k,h} = \{L_k^i : \ln L_k^i \geq \ln H + \frac{1}{2} \sigma_i^2(t_k) h - \sigma_i(t_k) \sqrt{h}\}, \quad (4.10)$$

i.e., the condition for $\ln L_{k+1}^i$ to be inside the domain G is

$$\ln L_k^i < \ln H + \frac{1}{2}\sigma_i^2(t_k)h - \sigma_i(t_k)\sqrt{h}, \quad (4.11)$$

and the corresponding λ_k for Algorithm 2.1 is so that

$$\lambda_k\sqrt{h} = -\frac{1}{2}\sigma_i^2(t_k)h + \sigma_i(t_k)\sqrt{h}. \quad (4.12)$$

We note that instead of (4.10) and (4.12) we could take the wider boundary zone $S_{t_k,h} = \{L_k^i : \ln L_k^i \geq \ln H - \sigma_i(t_k)\sqrt{h}\}$ and correspondingly $\lambda_k = \sigma_i(t_k)$. A wider boundary zone usually leads to a bigger numerical integration error. In this example we cannot take a boundary zone narrower than $S_{t_k,h}$ in (4.10) because it would not ensure that the chain $\ln L_k^i$ belongs to \bar{G} .

To realize Algorithm 2.1, we follow the random walk generated by (4.8) and at each time t_k , we check whether at the next step L_{k+1}^i cannot cross the barrier H , i.e., we check whether the condition (4.11) holds. If it does, we perform (4.8)-(4.9) to find $\ln L_{k+1}^i, Z_{k+1}$. Otherwise, L_k^i has reached the boundary zone $S_{t_k,h}$, where we make the auxiliary step: we either stop the chain at $\ln H$ with probability p :

$$p = \frac{\lambda_k\sqrt{h}}{\ln H - \ln L_k^i + \lambda_k\sqrt{h}}$$

or we kick the current position of the random walk $\ln L_k^i$ back into the domain to the position $\ln L_k^i - \lambda_k\sqrt{h}$ with probability $1-p$ and then carry out (4.8)-(4.9) to find $\ln L_{k+1}^i, Z_{k+1}$. If $k+1 = M$, we stop, otherwise we continue with the algorithm. The outcome of simulating each trajectory is a point $(t_{\varkappa}, \ln L_{\varkappa}^i, Z_{\varkappa})$.

To realize Algorithm 2.2, we also follow the random walk generated by (4.8), and at each time t_k , we check whether the condition (4.11) holds. If it does not, L_k^i has reached the boundary zone $S_{t_k,h}$ and we stop the chain at $\ln H$. If it does, we perform (4.8)-(4.9) to find $\ln L_{k+1}^i, Z_{k+1}$. If $k+1 = M$, we stop, otherwise we continue with the algorithm. The outcome of simulating each trajectory is again a point $(t_{\varkappa}, \ln L_{\varkappa}^i, Z_{\varkappa})$.

In the experiments we evaluate the expectation

$$\begin{aligned} \tilde{V}_{caplet}(0) &= E^{Q^{T_i+1}} \left[(L^i(T_i) - K)_+ \chi(\theta > T_i) + Z(\tau) \right] \\ &\approx E^{Q^{T_i+1}} \left[(\exp(\ln L_{\varkappa}^i) - K)_+ \chi(\varkappa = M) + Z_{\varkappa} \right]. \end{aligned} \quad (4.13)$$

The approximate equality in (4.13) is related to the bias due to the numerical approximation. The expectation on the right-hand side is realized via the Monte Carlo technique.

4.2 Numerical results

Here we present some results of numerical tests of Algorithms 2.1 and 2.2 for pricing the barrier caplet (4.13). We use the following parameters in the experiments: $i = 9$, $K = 1\%$, $H = 28\%$, $\delta = 1$, $L^9(0) = 13\%$. The volatility $\sigma_i(t)$ is

assumed to be constant at 25% . The exact caplet price with these parameters evaluated by (4.3) is 6.57%. In the experiments we did 10^6 Monte Carlo runs. The results are presented in Figure 4.1. We see that Algorithm 2.1 is much more accurate than Algorithm 2.2. We also observe “oscillating” convergence which is typical for binary tree methods [5].

Figure 4.1: *Barrier caplet price:* Comparision of the results of numerical experiments for the Algorithm 2.1 (Algorithm $O(h)$) and Algorithm 2.2 (Algorithm $O(\sqrt{h})$) and the exact caplet price (solid line) evaluated for $i = 9$, $K = 1\%$, $H = 28\%$, $\delta = 1$, $L^9(0) = 13\%$, $\sigma_i(t) = 25\%$.

Let us also remark on the effect of variance reduction in these experiments. For instance, in Algorithm 2.1 for $h = 0.02$ we got the Monte Carlo error (i.e., half of the size of the confidence interval for corresponding estimator with probability 0.95) equal to 1.11×10^{-4} in the case of $F = 0$ and 1.55×10^{-5} in the case of the optimal F from (4.7) (i.e., 100 time speed-up in reaching the same level of the Monte Carlo error). The use of the optimal F does not result in zero Monte Carlo error due to the error of numerical integration.

5 Trigger swap

This example is devoted to evaluation of multi-asset barrier options with barriers on all or some of the correlated underlying assets. We consider a trigger swap as a specific case, though the considered approach can be used to value other multi-asset barrier options, for instance basket options, CDOs and n^{th} -to-default CDSs, and it can also be applied for options with single, double and time-dependent barriers.

A trigger swap is a swap on a floating reference rate that takes effect or terminates when some index rate hits a specified trigger level. Trigger swaps have a number of variations [4, 21]. Here we consider a knock-in version of a payer trigger swap with a fixed rate K whose barrier is continuously monitored. The index and reference rate both coincide with a LIBOR rate. For given trigger levels H^0, \dots, H^{N-1} associated with the LIBOR rates $L^0(t), \dots, L^{N-1}(t)$, the structure of the swap under consideration is expressed as follows. Once one of the continuously monitored LIBOR rate $L^0(t), \dots, L^{N-1}(t)$ for the first time hits the corresponding trigger level H^0, \dots, H^{N-1} from below, the contract holder enters into the payer swap starting at next tenor date for the remaining time to the last tenor T_{N-1} . More specifically, let θ be the first exit time of $L^0(s), \dots, L^{N-1}(s)$, $s \geq 0$, from the domain $G = (0, H^0) \times \dots \times (0, H^{N-1})$, τ be the first exit time of the space-time diffusion $(s, L^0(s), \dots, L^{N-1}(s))$ from the domain $Q = [0, T_{N-1}) \times G$ (clearly $\tau = \theta \wedge T_{N-1}$), and $T_{\varrho(\tau)}$ be the closest tenor date T_i to τ from the right, i.e., $\varrho(t)$ is defined as

$$\varrho(t) = \min \{i, i = 0, 1, \dots, N-1 : t \leq T_i\}.$$

If $\theta \leq T_{N-1}$, then at a tenor date $T_{\varrho(\tau)}$ the contract holder enters into the contract according to which the holder pays fixed payments of δK and receives floating payments of $\delta L^{i-1}(T_{i-1})$ at the coupon dates T_i , $i = \varrho(\tau) + 1, \dots, N$; otherwise the contract expires worthless.

The value of this trigger swap at time $t = 0$ under the forward measure \mathbb{Q}^{T_N} is given by

$$\begin{aligned} V_{trswap}(0) &= P(0, T_N) E^{\mathbb{Q}^{T_N}} \left[\frac{1}{P(T_{\varrho(\tau)}, T_N)} \right. \\ &\quad \times \left(1 - P(T_{\varrho(\tau)}, T_N) - K\delta \sum_{i=\varrho(\tau)+1}^N P(T_{\varrho(\tau)}, T_i) \right) \chi(\theta \leq T_{N-1}) \left. \right], \end{aligned} \quad (5.1)$$

or in terms of the LIBOR rates

$$\begin{aligned} V_{trswap}(0) &= P(0, T_N) E^{\mathbb{Q}^{T_N}} \left[\left(\prod_{j=\varrho(\tau)}^{N-1} (1 + \delta L^j(T_{\varrho(\tau)})) \right. \right. \\ &\quad \left. \left. - K\delta \sum_{i=\varrho(\tau)+1}^N \prod_{j=i}^{N-1} (1 + \delta L^j(T_{\varrho(\tau)})) - 1 \right) \chi(\theta \leq T_{N-1}) \right]. \end{aligned} \quad (5.2)$$

In order to price this contract by the Monte Carlo technique, we need to generate paths for the vector $L^{\varrho(t)}(t), \dots, L^{N-1}(t)$. This means that the size of the vector of the LIBOR rates which we need to simulate decreases over time. The dynamics of $L^i(t)$ under Q^{T_N} are described by the SDEs (cf. (3.3)):

$$\frac{dL^i(t)}{L^i(t)} = -\sigma_i(t) \sum_{j=i+1}^{N-1} \frac{\delta L^j(t)}{1 + \delta L^j(t)} \rho_{i,j} \sigma_j(t) dt + \sigma_i(t) dW_i^{T_N}(t), \quad i = \varrho(t), \dots, N-1. \quad (5.3)$$

5.1 Algorithm

Here we only apply Algorithm 2.1. For simplicity, we consider such set of tenor dates T_i and time steps h that T_i/h are integers.

As before, we simulate dynamics of the LIBOR rates $L^{\varrho(t)}(t), \dots, L^{N-1}(t)$ in log space according to the weak Euler scheme (cf. (2.6)):

$$\begin{aligned} \ln L_{k+1}^i &= \ln L_k^i - \sigma_i(t_k) h \sum_{j=i+1}^{N-1} \frac{\delta L_k^j}{1 + \delta L_k^j} \rho^{ij} \sigma_j(t_k) \\ &\quad - \frac{1}{2} (\sigma_i(t_k))^2 h + \sigma_i(t_k) \sqrt{h} \sum_{j=\varrho_k}^N U_{i,j} \xi_{j,k+1}, \\ i &= \varrho_{k+1}, \dots, N-1, \end{aligned} \quad (5.4)$$

where $\xi_{j,k}$ are mutually independent random variables distributed by the law $P(\xi = \pm 1) = 1/2$ and $\varrho_k =: \varrho(t_k)$.

The algorithm for the considered trigger swap proceeds as follows. Let $L_k = (L_k^{\varrho_k}, \dots, L_k^{N-1})^\top$. Denote by \varkappa the first exit time of (t_k, L_k) from Q . Let $M = T_{N-1}/h$. Suppose by time step k none of the rates $L_k^{\varrho_k}, \dots, L_k^{N-1}$ have crossed their barriers $H^{\alpha_k}, \dots, H^{N-1}$, i.e., $\chi(\varkappa \leq k) = 0$. Then we evaluate whether at the next time step $k+1$ the event $\varkappa = k+1$ might be realized. One can see that the rate L_{k+1}^i , $i = \varrho_{k+1}, \dots, N-1$, computed via (5.4) will be below the barrier H^i , i.e. inside the domain G , if the following is true

$$\ln L_k^i < \ln H^i - \lambda_k \sqrt{h}, \quad (5.5)$$

where

$$\lambda_k = \sigma_{Max} \sqrt{N - \varrho_k}$$

and $\sigma_{Max} = \max_{j,k} \sigma_j(t_k)$.

If (5.5) is satisfied at time t_k for all rates $L_k^{\varrho_{k+1}}, \dots, L_k^{N-1}$ then we move to step t_{k+1} , evaluate $\ln L_{k+1}^i$, $i = \varrho_{k+1}, \dots, N-1$, according to (5.4) and continue with the algorithm unless $k+1 = M$ (in this case the trigger swap expires worthless).

The case when the condition (5.5) does not hold for a single i implies that the point L_k is in the boundary zone $S_{t_k, h}$ and is near the barrier H^i . Then we

either assign $\varkappa = k$, $\ln L_{\varkappa}^i = \ln H^i$, $\ln L_{\varkappa}^j = \ln L_k^j$ for $j \neq i$, $T_{\varepsilon_{\varkappa}} = T_{\varepsilon_k}$ with probability

$$p^i = \frac{\lambda_k \sqrt{h}}{\ln H^i - \ln L_k^i + \lambda_k \sqrt{h}} \quad (5.6)$$

and carry on with simulating $\ln L_{k+1}^i$, $i = \varrho_{k+1}, \dots, N-1$, according to (5.4) starting from $\ln L_{\varkappa}$ until time $T_{\varepsilon_{\varkappa}} = \min \{T_i : t_{\varkappa} \leq T_i, i = 0, 1, \dots, N-1\}$ (the barriers are “removed” in simulating the remaining part of this trajectory); or we jump outside the boundary zone $S_{t_k, h}$ by changing the i^{th} component of $\ln L_k$ from $\ln L_k^i$ to $\ln L_k^i - \lambda_k \sqrt{h}$ with probability $1 - p^i$, perform the usual step according to (5.4) and continue with the algorithm unless $k+1 = M$ (in this case the trigger swap expires worthless).

We note that in comparison with the original formulation of Algorithm 2.1 here we do not stop the chain L_k at its first exit time from the space domain G . Instead, when the barrier is hit, we find the trigger tenor date $T_{\varepsilon_{\varkappa}}$, and if $t_{\varkappa} < T_{N-1}$, we continue the simulation according to (5.4) until $T_{\varepsilon_{\varkappa}}$ to get the required LIBOR rates $L^j(T_{\varepsilon_{\varkappa}})$ in (5.2).

Now, we discuss the case when the condition (5.5) does not hold for more than one i (i.e., the random walk has reached a corner of the domain G). In this case, the algorithm proceeds as follows. Let us denote by $\ell = \{l_1, \dots, l_n\}$ the set of tenor dates corresponding to the LIBOR rates for which (5.5) is violated. First, we select the rate from the set $\{\ln L_k^{l_1}, \dots, \ln L_k^{l_n}\}$, which is the closest to its boundary, i.e. l_j such that $\ln H^{l_j} - \ln L_k^{l_j}$ is minimum over $j = 1, \dots, n$. Then, we repeat the procedure which is given above for the single i with the following difference. If $\ln L_k^i$ jumps from the boundary to $\ln L_k^i - \lambda_k \sqrt{h}$, we find the second closest rate from the set $\{\ln L_k^{l_1}, \dots, \ln L_k^{l_n}\}$ and as before repeat for this point the routine we have presented for the single i . We follow this procedure in the outlined fashion until either the set ℓ is empty or for some l_j we reach the boundary and assign $\ln L_k^{l_j} = \ln H^{l_j}$.

The outcome of simulating each trajectory is the payer swap starting tenor date $T_{\varepsilon_{\varkappa}}$, the stopping time \varkappa and the point $\ln L_{\eta}$ with $\eta = T_{\varepsilon_{\varkappa}}/h$, which are used for evaluating the trigger swap:

$$V_{trswap}(0) \approx P(0, T_N) E^{Q^{T_N}} \left[\left(\prod_{j=\varrho_{\varkappa}}^{N-1} (1 + \delta L_{\eta}^j) - K \delta \sum_{i=\varrho_{\varkappa}+1}^N \prod_{j=i}^{N-1} (1 + \delta L_{\eta}^j) - 1 \right) \chi(\varkappa < M) \right]$$

with the expectation simulated by the Monte Carlo technique. We present the pseudocode for simulating a single trajectory based on the algorithm we described above.

Algorithm 5.1 Pseudocode for simulating a single trajectory in pricing the barrier trigger swap

```

SET  $M$  to  $T_{N-1}/h$ ,  $k$  to 1,  $\kappa$  to  $M$ 
WHILE  $k < M$ 
  IF  $\kappa > k$ 
    FOR  $j = 0$  to  $N - 1$ ,
      IF (5.5) for  $j$  is false
        calculate  $p^j$  by (5.6)
        form array  $p$  of  $p^j$ 
      ENDIF
    ENDFOR
    sort  $p$  in descending order
    FOR  $n = 1$  to  $length(p)$ 
      generate  $u \sim Unif[0.1]$ ;
      IF  $u < p(n)$ 
        SET  $\kappa$  to  $k$ 
        SET  $\ln L_k^i$  to  $\ln H^i$ 
        SET  $M$  to  $T_{e_\kappa}/h$ 
        BREAK
      ELSE
        SET  $\ln L_k^i$  to  $\ln L_k^i - \lambda_k \sqrt{h}$ 
      ENDIF
    ENDFOR
  ENDIF
  Evaluate  $\ln L_{k+1}$  by (5.4)
  Increase  $k$  by 1
ENDWHILE

```

5.2 Numerical results

Let us present results of numerical experiments we performed for pricing a trigger swap with Algorithm 6.1. The parameters chosen for the experiments are $T^0 = 5$, $T^* = 16$, $K = 0.01$, $H = 0.13$, $\delta = 1$, $\beta = 0.2$. The initial LIBOR rate curve is assumed to be flat at 4% and the volatility $\sigma_i(t)$ is set to be constant at 20%. In the simulations we run 10^6 Monte Carlo paths.

Since the closed-form formula for trigger swap (5.2) is not available, we found the reference trigger swap price by evaluating the price using Algorithm 5.1 with $h = 0.01$ and the number of Monte Carlo runs 10^6 . This reference price is 5.46×10^{-2} with the Monte Carlo error 5.50×10^{-4} , which gives half of the size of the confidence interval for the corresponding estimator with probability 0.95.

The results of the experiments with Algorithm 5.1 are presented in Table 5.1. In the table, the values before “ \pm ” are estimates of the bias computed as the difference between the reference price and its sampled approximation, while the values after “ \pm ” give half of the size of the confidence interval for the corresponding estimator with probability 0.95. The “mean exit time” is the average time for trajectories (t_k, L_k) to leave the space-time domain Q . The experimentally observed convergence rate for Algorithm 5.1 is in agreement with

the theoretical first order convergence in h (though we note that the convergence theorem in [15, 16] is proved under restrictive regularity conditions and the payoff of the trigger swap and the boundary of the space domain G do not satisfy these conditions).

Table 5.1: *Performance of Algorithm 5.1 for the trigger swap.*

h	error	mean exit time
0.25	$2.22 \times 10^{-2} \pm 6.39 \times 10^{-4}$	12.51
0.2	$1.85 \times 10^{-2} \pm 6.26 \times 10^{-4}$	12.61
0.125	$1.17 \times 10^{-2} \pm 6.01 \times 10^{-4}$	12.78
0.1	$9.56 \times 10^{-3} \pm 5.92 \times 10^{-4}$	12.83
0.0625	$6.03 \times 10^{-3} \pm 5.78 \times 10^{-4}$	12.92
0.05	$4.67 \times 10^{-3} \pm 5.72 \times 10^{-4}$	12.95

6 Barrier swaption

In this section we consider Monte Carlo evaluation of a knock-out swaption under the LMM. We use the knock-out swaption as a guide in our exposition, its treatment is rather general and it can be used to value different barrier options, where the underlying and barrier can be expressed as functionals of some diffusion process.

A European payer (receiver) swaption is an option that gives its holder a right, but not an obligation, to enter a payer (receiver) swap at a future date at a given fixed rate K . Usually, the swaption maturity coincides with the first reset date T_0 of the underlying swap. The underlying swap length $T_N - T_0$ is called the tenor of the swaption.

Without loss of generality, we concentrate on a knock-out receiver swaption with the first reset date T_0 . A knock-out swaption has the structure as a standard swaption except that if the underlying swap rate is above a barrier level R_{up} at any time before T_0 then the swaption expires worthless. The price of the knock-out swaption at time $t = 0$ under the forward measure \mathbb{Q}^{T_0} is given by:

$$V_{swaption}(0) = P(0, T_0) E^{\mathbb{Q}^{T_0}} \left[\delta (R_{swap}(T_0) - K)_+ \sum_{j=1}^N P(T_0, T_j) \chi(\theta > T_0) \right], \quad (6.1)$$

where θ is the first exit time of the process $R_{swap}(s)$, $s \geq 0$, from the interval $(0, R_{up})$. The swap rate $R_{swap}(s)$ can be expressed in terms of the spanning LIBOR rates as

$$R_{swap}(s) = \frac{1 - 1 / \prod_{j=0}^{N-1} (1 + \delta L^j(s))}{\delta \sum_{i=0}^{N-1} 1 / \prod_{j=0}^i (1 + \delta L^j(s))}. \quad (6.2)$$

The bond prices $P(T_0, T_j)$ can also be expressed via LIBOR rates (see (3.1)).

Also, let τ be the first exit time of the space-time process $(s, R_{swap}(s))$ from the domain $D = [0, T_0) \times (0, R_{up})$ (obviously, $\tau = \theta \wedge T_0$).

We note that expression (6.1) depends on the joint distribution of the forward rates $L^0(T_0), \dots, L^{N-1}(T_0)$. The LMM dynamics of LIBOR rates under Q^{T_0} are given by (cf. (3.3)):

$$\frac{dL^i(t)}{L^i(t)} = \sigma_i(t) \sum_{j=0}^i \frac{\delta L^j(t)}{1 + \delta L^j(t)} \rho_{i,j} \sigma_j(t) dt + \sigma_i(t) dW_i^{T_0}(t), \quad i = 0, \dots, N-1. \quad (6.3)$$

In this example we deal with pricing the barrier swaption (6.1) expressed in terms of the spanning LIBOR rates with dynamics in the form of (6.3). This means that we consider this problem in the coordinate system of the spanning LIBOR rates and the barrier is given as an implicit surface in the LIBOR coordinates. We also note that there is the space domain G in the phase space of the SDEs (6.3) corresponding to the interval $(0, R_{up})$ on the swap-rate semi-line. As usual, the corresponding space-time domain $Q := [0, T_0) \times G$.

For test purposes, let us introduce an analytical approximation for the barrier swaption. To this end, we note that under the Swap Market Model (SMM, see details in [4, 18, 21]) the barrier swaption pricing problem admits the closed-form solution (cf. (4.3))

$$\begin{aligned} V_{swaption}(0) = & \delta \sum_{j=1}^N P(0, T_j) \left\{ R_{swap}(0) [\Phi(\delta_+(R_{swap}(0)/K, v_{R_{swap}})) \right. \\ & - \Phi(\delta_+(R_{swap}(0)/R_{up}, v_{R_{swap}}))] \\ & - K [\Phi(\delta_-(R_{swap}(0)/K, v_{R_{swap}})) - \Phi(\delta_-(R_{swap}(0)/R_{up}, v_{R_{swap}}))] \\ & - H [\Phi(\delta_+(R_{up}^2/(K R_{swap}(0)), v_{R_{swap}})) - \Phi(\delta_+(R_{up}/R_{swap}(0), v_{R_{swap}}))] \\ & + K R_{swap}(0) \Phi(\delta_-(R_{up}^2/(K R_{swap}(0)), v_{R_{swap}})/R_{up}) \\ & \left. - \Phi(\delta_-(R_{up}/R_{swap}(0), v_{R_{swap}}))] \right\}, \end{aligned} \quad (6.4)$$

where δ_{\pm} are from (4.4),

$$v_{R_{swap}}^2 = \int_0^{T_i} (\sigma_{R_{swap}}(s))^2 ds,$$

and $\sigma_{R_{swap}}(s)$ is the instantaneous volatility of the log-normal dynamics of the swap rate.

Using Rebonato's formula [4, p. 283], we can approximately compute the "approximate" volatility $v_{R_{swap}}^{LMM}$ for the LMM analogous to the volatility $v_{R_{swap}}$ in the SMM entering (6.4) as

$$v_{R_{swap}}^{LMM} = \sum_{i,j=0}^{N-1} \frac{\omega_i(0)\omega_j(0)L^i(0)L^j(0)\rho_{ij}}{(R_{swap}(0))^2} \int_0^{T_0} \sigma_i(s)\sigma_j(s)ds, \quad (6.5)$$

where

$$\omega_i(0) = \frac{1 - 1/\prod_{j=0}^{i-1} (1 + \delta L^j(0))}{\delta \sum_{k=0}^{N-1} 1/\prod_{j=0}^k (1 + \delta L^j(0))}.$$

The quantity $v_{R_{swap}}^{LMM}$ can be used as a proxy for $v_{R_{swap}}$ in (6.4) to compute approximated barrier swaption prices under LMM. We will check in our numerical experiments whether an approximation obtained by our algorithm is consistent with this analytical approximation.

6.1 Algorithm

Here we exploit Algorithm 2.1. We choose a time step $h > 0$ so that $M = T_0/h$ is an integer. Again inside the domain G we use the weak Euler scheme to simulate trajectories of the log LIBOR rates (6.3):

$$\begin{aligned} \ln L_{k+1}^i &= \ln L_k^i + \sigma_i(t_k) \sum_{j=0}^i \frac{\delta L_k^j}{1 + \delta L_k^j} \rho_{ij} \sigma_j(t_k) h \\ &\quad - \frac{1}{2} (\sigma_i(t_k))^2 h + \sigma_i(t_k) \sqrt{h} \sum_{j=0}^{N-1} U_{i,j} \xi_{j,k+1}, \\ i &= 0, \dots, N-1, \end{aligned} \quad (6.6)$$

where $\xi_{j,k}$ are mutually independent random variables distributed by the law $P(\xi = \pm 1) = 1/2$.

For a fixed t_k , we denote by $\ln L_k$ the point with coordinates $\ln L_k^0, \ln L_k^1, \dots, \ln L_k^{N-1}$, i.e. $\ln L_k = (\ln L_k^0, \ln L_k^1, \dots, \ln L_k^{N-1})^\top$. As before, we follow the random walk constructed by (6.6) until we reach the boundary zone $S_{t_k,h}$. Algorithmically, it implies that we implement a check at each step whether the current position of the random walk is in the boundary zone $S_{t_k,h}$. More precisely, we evaluate at time t_k whether the current position $\ln L_k$ is such that the maximum increment from point $\ln L_k$ according to all possible realizations of (6.6) at the next time level t_{k+1} results in the state of the random walk below the barrier, i.e. in the domain G .

Introduce

$$\ln L_{k,Max} = \max_i \ln L_k^i$$

and

$$\ln \hat{L}_{k+1} = \ln L_{k,Max} + \sigma_{Max}^2 h N - \frac{1}{2} \sigma_{Max}^2 h + \sigma_{Max} \sqrt{h N}, \quad (6.7)$$

where $\sigma_{Max} = \max_{i,k} \sigma_i(t_k)$. Using the fact that

$$R_{swap}(\hat{L}_{k+1}, \dots, \hat{L}_{k+1}) = \hat{L}_{k+1},$$

one can see that the current position of the random walk $\ln L_k$ is inside the domain G if the following condition is satisfied

$$\ln \hat{L}_{k+1} < \ln R_{up}. \quad (6.8)$$

Algorithmically, we do the following. If condition (6.8) is true, we evaluate the next position of the random walk at t_{k+1} according to (6.6) and continue further with the algorithm unless $k+1 = M$ (i.e., we have reached the maturity time T_0 of the swaption).

We note the condition (6.8) is computationally easy to evaluate but it is rather rough. Once this condition fails, we check a finer but computationally more expensive condition based on the maximum increments from each of $L^i(t_k)$ towards the boundary:

$$R_{swap}(L_k^0(1 + \sigma_0(t_k)\sigma_{Max,k}h + \sigma_0(t_k)\sqrt{Nh}), L_k^1(1 + 2\sigma_1(t_k)\sigma_{Max,k}h + \sigma_1(t_k)\sqrt{(N-1)h}), \dots, L_k^{N-1}(1 + N\sigma_{N-1}(t_k)\sigma_{Max,k}h + \sigma_{N-1}(t_k)\sqrt{h})) < R_{up}, \quad (6.9)$$

where $\sigma_{Max,k} = \max_j \sigma_j(t_k)$. If condition (6.9) holds, we again carry on to the next time step t_{k+1} using (6.6) and continue further with the algorithm unless $k+1 = M$.

If both conditions (6.8) and (6.9) fail, the random walk has reached the boundary zone $S_{t_k,h}$, where as before we apply the different procedure which require us to find the projection $\ln L_k^\pi := (L_k^{\pi,0}, L_k^{\pi,1}, \dots, L_k^{\pi,N-1})^\top$ of the current position $\ln L_k$ on the boundary given as the implicit function of the spanning LIBOR rates:

$$\ln R_{swap}(t_k) = \ln R_{up}. \quad (6.10)$$

For completeness of the exposition, let us discuss how the projection $\ln L_k^\pi$ can be simulated before we return to the description of the algorithm. The problem of finding point $\ln L_k^\pi$ is equivalent to finding the minimum value of the function

$$|\ln L_k^\pi - \ln L_k|^2 = \left(\ln L_k^{\pi,0} - \ln L_k^0\right)^2 + \dots + \left(L_k^{\pi,N-1} - \ln L_k^{N-1}\right)^2 \quad (6.11)$$

subject to the constraint

$$\ln \left(\frac{\prod_{j=0}^{N-1} (1 + \delta L_k^{\pi,j}) - 1}{\delta \left(1 + \sum_{i=0}^{N-2} \prod_{j=i+1}^{N-1} (1 + \delta L_k^{\pi,j}) \right)} \right) = \ln R_{up}. \quad (6.12)$$

We regard $\ln L_k^{\pi,1}, \dots, \ln L_k^{\pi,N-1}$ as independent variable in the constraint equa-

tion (6.12) and write $\ln L_k^{\pi,0}$ as

$$\ln L_k^{\pi,0} = \ln \left(\frac{R_{up} \cdot \left(1 + \sum_{i=0}^{N-2} \prod_{j=i+1}^{N-1} (1 + \delta L_k^{\pi,j}) \right) + 1}{\prod_{j=1}^{N-1} (1 + \delta L_k^{\pi,j})} - \frac{1}{\delta} \right). \quad (6.13)$$

Hence the minimization problem is reduced to finding the point $\ln L_k^{\pi,1}, \dots, \ln L^{\pi,N-1}$ at which the function $|\ln L_k^{\pi} - \ln L_k|^2$ from (6.11) with $\ln L_k^{\pi,0}$ from (6.13) has its minimum value. This optimization problem can be solved using standard procedures, e.g. the MATLAB function “lsqnonlin()”.

Let us now continue with the description of the algorithm. When $\ln L_k$ is in the boundary zone, we either stop the chain at $\ln L_k^{\pi}$ with probability p :

$$p = \frac{\lambda\sqrt{h}}{|\ln L_k^{\pi} - \ln L_k| + \lambda\sqrt{h}}, \quad (6.14)$$

where

$$\lambda\sqrt{h} = \sqrt{N} \left(\sigma_{Max}^2 h N - \frac{1}{2} \sigma_{Max}^2 h + \sigma_{Max} \sqrt{hN} \right); \quad (6.15)$$

or we jump inside the domain Q to the point $\ln L_k + \lambda\sqrt{h} \frac{\ln L_k^{\pi} - \ln L_k}{|\ln L_k^{\pi} - \ln L_k|}$ with probability $1 - p$, apply the Euler step (6.6) to evaluate $\ln L_{k+1}$ and continue further with the algorithm unless $k + 1 = M$.

The outcome of simulating each trajectory is the point $(t_{\mathcal{X}}, \ln L_{\mathcal{X}})$. In Algorithm 6.1 we present the pseudocode for simulating a single trajectory based on the algorithm we described above.

6.2 Numerical results

We give some results for pricing a barrier swaption by Algorithm 6.1. We consider the barrier swaption with the initial LIBOR curve flat at 5%, constant volatility $\sigma_i(t)$ at 10% and the following parameters: $T_0 = 10$, $T^* = 20$, $K = 0.01$, $R_{up} = 0.075$, $\delta = 1$, $\beta = 0.1$. The simulations use 10^6 Monte Carlo runs.

The pricing problem for the barrier swaption (6.1) does not admit a closed-form solution. We used the barrier swaption price found with Algorithm 6.1 with $h = 0.01$ and 10^7 of Monte Carlo runs as the reference solution. This reference price is 0.15506 with the Monte Carlo error 1.42×10^{-4} , which gives half of the size of the confidence interval for the corresponding estimator with probability 0.95. The analytical approximation based on (6.4) and (6.5) yields the price of the barrier swaption 0.15556.

We present results of the experiments in Table 6.1. As in the previous section, the error column values before “ \pm ” are estimates of the bias computed using the reference price value and the values after “ \pm ” reflect the Monte Carlo error with probability 0.95. The “mean exit time” is the average time for approximate trajectories to exit the space-time domain Q . It is clear that the results demonstrate the expected first order of convergence.

Algorithm 6.1 Barrier swaption: Pseudocode for simulating a single trajectory

```
FOR  $k = 1$  to  $M$ 
  calculate  $\ln \hat{L}_{k+1}$  by (6.7)
  IF (6.8) is true: calculate  $\ln L_{k+1}$  by (6.6)
  ELSE
    IF (6.9) is true: calculate  $\ln L_{k+1}$  by (6.6)
    ELSE
      solve the minimisation problem (6.11) with  $\ln L_k^{\pi,0}$  from (6.13)
      generate  $u \sim \text{Unif}[0, 1]$ 
      calculate probability  $p$  by (6.14)
      IF  $u < p$ 
        break
      ELSE
        SET  $\ln L_k$  to  $\ln L_k + \lambda \sqrt{h} \frac{\overrightarrow{\ln L_k^\pi \ln L_k}}{|\ln L_k^\pi - \ln L_k|}$ 
        calculate  $\ln L_{k+1}$  by (6.6)
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDFOR
```

Table 6.1: *Performance of Algorithm 6.1 for the barrier swaption.*

h	error	mean exit time
0.25	$1.01 \times 10^{-2} \pm 4.33 \times 10^{-4}$	9.36
0.2	$8.08 \times 10^{-3} \pm 4.37 \times 10^{-4}$	9.40
0.125	$5.15 \times 10^{-3} \pm 4.42 \times 10^{-4}$	9.46
0.1	$4.15 \times 10^{-3} \pm 4.44 \times 10^{-4}$	9.48
0.0625	$2.58 \times 10^{-3} \pm 4.47 \times 10^{-4}$	9.51
0.03125	$1.03 \times 10^{-3} \pm 4.49 \times 10^{-4}$	9.54

Acknowledgment

MVT was partially supported by the Leverhulme Trust.

References

- [1] L. Andersen, R. Brotherton-Racliffe. Exact exotics. *Risk* **9** (1996), 85–89.
- [2] P. Baldi. Exact asymptotics for the probability of exit from a domain and applications to simulation. *Ann. Prob.* **23** (1995), 1644–1670.
- [3] D. R. Beaglehole, P.H. Dybvig, G. Zhou. Going to extremes: Correcting simulation bias in exotic option valuation. *Financial Analyst Journal*, January/February (1997), 62–68.

- [4] D. Brigo, F. Mercurio. *Interest Rate Models: Theory and Practice*. Springer, 2006.
- [5] E. Derman, I. Kani, D. Ergener, I. Bardhan. Enhanced numerical methods for options with barriers. *Quantitative Strategies Research Notes*, Goldman Sachs. May 1995.
- [6] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.
- [7] E. Gobet. Weak approximation of killed diffusion using Euler schemes. *Stoch. Process. Appl.* **87** (2000), 167–197.
- [8] E. Gobet. Advanced Monte Carlo methods for barrier and related exotic options. In *Mathematical Modeling and Numerical Methods in Finance*, Eds. A. Bensoussan, Q. Zhang, P. Ciarlet. Elsevier, 2009, 497–528.
- [9] P.E. Kloeden, E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.
- [10] N. Kunitomo, M. Ikeda. Pricing options with curved boundaries. *Math. Fin.* **4** (1992), 275–298.
- [11] G.N. Milstein. *Numerical Integration of Stochastic Differential Equations*. Kluwer Academic Publishers, 1995.
- [12] G.N. Milstein. Solution of the first boundary value problem for equations of parabolic type by means of the integration of stochastic differential equations. *Theor. Probab. Appl.*, **40** (1995), 556–563.
- [13] G.N. Milstein, N.F. Rybkina. An algorithm for random walks over small ellipsoids for solving the general Dirichlet problem. *J. Comp. Math. Math. Phys.*, **33** (1993), 631–647.
- [14] G.N. Milstein, J.G.M. Schoenmakers. Monte Carlo construction of hedging strategies against multi-asset European claims. *Stoch. Stoch. Rep.* **73** (2002), 125–157.
- [15] G.N. Milstein, M.V. Tretyakov. The simplest random walks for the Dirichlet problem. *Theory Prob. Appl.* **47** (2002), 53–68.
- [16] G.N. Milstein, M.V. Tretyakov. *Stochastic Numerics for Mathematical Physics*. Springer, 2004.
- [17] G.N. Milstein, M.V. Tretyakov. Practical variance reduction via regression for simulating diffusions. *SIAM J. Numer. Anal.* **47** (2009), 887–910.
- [18] R. Rebonato. *Interest-rate Option Models*. Wiley, 1998.
- [19] M. Rubinstein, E. Reiner. Breaking down the barriers. *Risk* **4** (1991), 28–35.

- [20] P. Shevchenko. Addressing the bias in Monte Carlo pricing of multiasset options with multiple barriers through discrete sampling. *J. Comp. Fin.* **6** (2003), 1–20.
- [21] J. Schoenmakers. *Robust Libor Modelling and Pricing of Derivative Products*. Chapman and Hall/CRC, 2005.