The valuation of exotic barrier options by contour bridge simulation

Pokpong Chirayukool
Warwick Business School
University of Warwick
Coventry CV4 7AL
United Kingdom
Pokpong.Chirayukool07@phd.wbs.ac.uk

Nick Webber*
Warwick Business School
University of Warwick
Coventry CV4 7AL
United Kingdom
Nick.Webber@wbs.ac.uk

August 18, 2012

Abstract

Options with non-constant continuous barriers occur in various contexts in finance. As well as existing as stand-alone OTC financial instruments, option features of this sort may be embedded in debt contracts, and can arise in commodity or credit applications.

In general, these options can be hard to value quickly and accurately. This article presents a general simulation method, the contour bridge method, that is capable of valuing options with difficult exotic barriers. Importantly the method is unbiased, and can be adapted to price a range of options, barrier and non-barrier, simultaneously; a significant advantage over methods that price a single option at a time.

The method can be applied with models where it is possible to sample from a suitable set of strictly increasing stopping times, specified as hitting times to amenable contours. These conditions are met not only when the underlying asset process is a geometric Brownian motion, but also for more general processes such as the variance gamma and normal inverse Gaussian processes.

Computational results demonstrate that compared to existing methods the contour bridge method offers significant variance reduction for both benchmark and exotic options.

Keywords: Barrier options; Monte Carlo simulation; variance reduction.

*Corresponding author. This version of the paper was presented at MathFinance 2011, Frankfurt, and at seminars at the University of Warwick and King’s College, London.


1 Introduction

Barrier options with flat, constant, barriers are standard exotic options. As well as being traded in their own right they, and more complex versions of them, also occur embedded in debt instruments, and arise in commodity and credit applications. The more complex versions are characterized by conditional or non-constant, time varying, barriers, making them hard to value. For the class of options with non-constant barriers we present a simulation method, the contour bridge method, based on the simulation of asset values at times determined by a sequence of strictly increasing stopping times, that can deliver substantial improvements in computation times over existing simulation methods. The contour bridge method is unbiased, and can be used to value barrier options with highly non-linear barriers, partial and discrete barrier options, and also non-barrier options.

The set of stopping times is specified as first passage times to a set of nested contours. The method requires that the first passage times, and also certain bridge distributions, can be sampled. These conditions can be met not only when the underlying process follows a geometric Brownian motion, but also when its process is more general, such as a variance-gamma or normal inverse Gaussian process.

Some existing methods for valuing barrier options can be used to price only a single option at a time. This can be very inefficient when a book of options is being valued. Even if a method can price a single option quite rapidly, its cost when applied to a book can be excessive. The contour bridge method can be adapted to price multiple options simultaneously.

The structure of the paper is as follows. Section 2 reviews current valuation methods for barrier options, with simulation methods reviewed in greater detail in section 3. In section 4 we present the contour bridge method and discuss enhancements that can facilitate further variance reduction. Numerical results are presented in section 5 where we find promising results for a set of exotic and benchmark options. Finally, section 6 concludes.

2 Valuation methods for barrier options

Existing methods for pricing barrier options include lattice methods, PDE methods, simulation methods, and analytical approximations and quadrature. The presence of one or more barriers causes particular difficulties for standard pricing methods. They have to be modified to achieve reasonable, unbiased, convergence. Generally, time-varying barriers present exceptional difficulties for fast and accurate valuation.

Valuation methods for standard barrier options Lattice methods have a long history of pricing barrier options. Techniques include shifting lattice nodes to lie on the barrier (Derman et al. (1995) [19], Dai and Lyuu (2007) [18]), incorporating conditional hitting probabilities into branching probabilities (Kuan and Webber (2003) [28], Barone-Adesi, Fusari and Theal (2008) [7]), refining the lattice branching when close to the barrier (Ahn, Figlewski and Gao (1999) [1]), et cetera.
Lattice methods can be effective when the number of state variables is small but are difficult to use with complex barriers and difficult processes. PDE-based methods can also be used when the number of state variables in the model is small. Explicit finite difference methods have been adapted by Boyle and Tian (1998) [11], implicit methods by Zvan, Vetzal and Forsyth (2000) [54] and Ndogmo and Ntwiga (2007) [40], and higher order Crank-Nicolson by Wade et al. (2007) [52]. Itken and Carr (2008) [25] price barrier options in the stochastic skew model of Carr and Wu (2004) [16] using LOD operator splitting methods. Finite element methods were used by Foufas and Larson (2004) [21].

PDE methods, like lattice methods, may be able to provide only a single option value at a time, a severe disadvantage if values for a book of options are required.

Other valuation methods include analytical approximations, often requiring numerical integration (Mijatović (2010) [38], Lo (1997) [36], and Thompson (2002) [51], amongst others), and Laplace transform methods can be used to price both double and single barrier options (see for instance Pelsser (2000) [42], Hulley and Platen (2007) [24], and Wang, Lui and Hsiao (2009) [53]).

Simulation methods are attractive in a number of situations, for instance when there are more than one or two state variables, or when the asset price process is not amenable to other methods.

Naive simulation methods for valuing barrier options can be slow, suffering from persistent bias leading to very low rates of convergence. In some cases it has been found possible to remove bias. For example, Baldi (1995) [2] and Beaglehole, Dybvig and Zhou (1997) [8] show how simulation bias can be removed when a certain conditional hitting time probability can be computed. The approach was extended to multi-asset options by Shevchenko (2003) [50], and approximate formulae for time dependent barriers were obtained by Baldi, Caramellino and Iovino (1999) [4]. However the use of these techniques is limited to cases where the conditional hitting time probability is either known or can be approximated with sufficient accuracy. This limits their range of application.

For a simulation method to be useful in practical applications it is necessary for variance reduction methods to be employed. Standard methods include stratified sampling, the use of control variates, and importance sampling. For instance importance sampling was used by Glasserman and Staum (2010) [22] when the underlying asset follows a geometric Brownian motion, and Joshi and Leung (2007) [26] use importance sampling with the Merton jump-diffusion process. We do not discuss the application of these variance reduction methods in conjunction with contour bridge sampling. On the whole they can be used with equal effectiveness by both standard simulation methods and the contour bridge method, so that the relative advantage of the contour bridge method persists when these variance reduction methods are incorporated.

**Valuation with non-GBM processes** Most authors consider only cases where the underlying asset follows a geometric Brownian motion (GBM). General Lévy processes are considered by Carr and Hirs (2007) [15], who obtain results with a PIDE method, and Boyarchenko and Levendorskii (2009) [10] who exploit Carr’s randomization approximation (Carr (1998) [14]) applied to barrier options. Joshi and Leung (2010) [26] and Metwally and Atiya (2002) [37] price barrier options when the asset value follows a Merton jump-diffusion process.


The contour bridge method can be used with both the variance gamma and normal inverse Gaussian processes as relevant stopping times can be simulated in these cases.

Options with non-constant barriers A number of authors have investigated non-constant barrier options. Exponential-affine barriers are relatively tractable and have been investigated by various authors, starting with Kunitomo and Ikeda (1992) [29]. Costabile (2002) [17] prices them on a binomial lattice. Options with linear barriers are priced by Rogers and Zane (1997) [48]. Rogers and Stapleton (1998) [47] use a modified binomial lattice to price constant and exponential-affine barrier options. Morimoto and Takahashi (2002) [39] consider barriers that vary with \( \sqrt{T} \). A piece-wise linear approximation to time-dependent barriers was used by Novikov, Frishling and Kordzakhia (2003) [41]. Partial barrier options where a (constant) barrier applies only over part of an option’s life were investigated by Hui (1997) [23].

Rogers and Zane (1997) [48] transform time-dependent barriers into constant barriers, replacing the process for the underlying asset by a process with time varying parameters. Approximate methods for models with time dependent parameters can then be applied.


A credit application with stochastic barriers was investigated by Kijima and Suzuki (2007) [27].

Before presenting the contour bridge simulation method we review in more detail existing standard simulation methods.

3 Simulation methods for barrier options

Consider an exotic barrier option maturing at a fixed time \( T \), with barrier \( b(t) \), \( t \in [0, T] \), when the initial asset value is \( S_0 \). Write \( \tau \) for the first hitting time of \( S = (S_t)_{t \geq 0} \) to the option barrier and let \( I = I_{\tau \leq T} \), where \( I \) is the indicator function, be the Boolean variable specifying whether or not the barrier is hit before time \( T \).

Let \( \omega \) be a continuous time sample path for the asset value. When a barrier option has at most a single payoff time write \( H(\omega) \) for the payoff to the option
along the path $\omega$ and $t(\omega)$ for the time at which the payoff is received. Usually the payoff is received either at the hitting time $\tau$ or at the maturity date of the option.

Barrier options fall into two main groups; those in which the payoff $H(\omega)$ or payoff time $t(\omega)$ depend directly on $\tau$, and those where they depend on $\tau$ only through $\iota$. We call the first group $\tau$-options and the second $\iota$-options. An example of a $\tau$-option is an option paying a rebate at the hitting time; an $\iota$-option a vanilla knock-in or knock-out barrier option. An $\iota$-option will often require the value of a further state variable, such as the value $S_T$ of the underlying asset at time $T$, to calculate its payoff. If the payoff is determined solely by the value of $\iota$, such as a knock-out rebate paying off at time $T$, we call the option a bare-$\iota$ option.

In a structural model of default, the default time is the first hitting time of an asset value to a some barrier. A bond behaves like a bare-$\iota$ option. In reduced form credit models default may be modelled as occurring when a count-down process hits a barrier level, so bond prices again resemble bare-$\iota$-options.

Throughout the article we suppose for simplicity that $S_0 > b(0)$ so that all options are ‘down’-style. Applications to ‘up’-style options, where $S_0 < b(0)$, follow by symmetry.

In this section we suppose that asset values are generated at a fixed set of times $\mathbf{Y} = \{t_i\}_{i=0,\ldots,N}$, with $t_0 = 0$, $t_N = T$, and $t_i < t_{i+1}$ for all $i$. Write $b^i = b(t_i)$ for the barrier value at time $t_i$ and $S^i_j$ for the simulated value at time $t_i$ on the $j$th sample path, with $S^0_j \equiv S_0$. Denote the $j$th sample path by $S_j = \{S^i_j\}_{i=0,\ldots,N}$.

We briefly sketch the Dirichlet and the Brownian bridge Monte Carlo methods. The contour bridge method extends both of these standard methods.

**Dirichlet Monte Carlo** A plain forward-evolution path-based Monte Carlo method simulates $M$ sample paths for the underlying asset, $\{S_j\}_{j=1,\ldots,M}$. The barrier is deemed to have been hit at time $t_i$ if $S^i_j < b^i$ but $S^{i-1}_j > b^{i-1}$, and the plain method hitting time, $\tau^P_j$, is set to be

$$\tau^P_j = \min \left\{ t_i \in \mathbf{Y} \mid S^i_j \leq b^i \right\}. \quad (1)$$

The Monte Carlo estimate $\hat{c}_\iota$ of the option value is

$$\hat{c}_\iota = \frac{1}{M} \sum_{j=1}^{M} e^{-r(t(S_j)-t)} H(S_j) \quad (2)$$

where $H(S_j)$ and $t(S_j)$ are the discrete approximation to $H(\omega)$ and $t(\omega)$ computed by the method from $S_j$.

The plain method is prone to severe simulation bias: hitting times are constrained to lie in $\mathbf{Y}$, and if both $S^i_j$ and $S^{i+1}_j$ lie close to, but above, the barrier, then it is possible that the barrier was hit in the interval $[t_i, t_{i+1}]$, undetected by the method. Consequently the method tends to under-price continuous knock-in barrier options and over-price continuous knock-out options. This is referred to as simulation bias.

The Dirichlet method (Beaglehole, Dybvig and Zhou (1997) [8]) corrects for simulation bias. On the $j$th sample path, when $S^i_j$ and $S^{i+1}_j$ lie above the
barrier, let \( p_j^i = \Pr \{ t_i < \tau < t_{i+1} | S_j^i, S_j^{i+1} \} \) be the hitting probability in the interval \([t_i, t_{i+1}]\) conditional on the process taking values \( S_j^i \) and \( S_j^{i+1} \) at times \( t_i \) and \( t_{i+1} \). If \( S_j^i > b^i \) for all \( i \), then \( p_j = 1 - \prod_{i=0}^{N-1} (1 - p_j^i) \) is the probability that the barrier was hit in between simulated times on the \( j \)th path.

When \( p_j^i \) can be computed and sampled, for instance when \( S \) is a geometric Brownian motion and \( b(t) \equiv b \) is a constant barrier, then \( p_j^i \) can be used to simulate the intra-period hitting time, or \( p_j \) to weight the payoff, to correct for simulation bias. This is a very effective mechanism for removing simulation bias.

When \( p_j^i \) can not be computed, perhaps because the barrier is not constant, or the asset process too difficult, it may be possible to approximate \( p_j^i \). For instance a time varying barrier \( b(t) \) might be approximated as piece-wise constant (or when the underlying asset process is geometric Brownian motion, piece-wise exponential-affine) over the intervals \([t_i, t_{i+1}]\).

Brownian bridge Monte Carlo  A Monte Carlo method with stratified sampling may generate a path \((S^0, \ldots, S^N)\) at times \( t_i \in \Upsilon \) by first constructing a value \( S^N \) for time \( t_N \) and then iteratively filling up the path in the order \( S^{[N/2]} \), \( S^{[N/4]} \), \( S^{[N/8]} \), \( S^{[3N/8]} \), \( S^{[3N/4]} \), \( S^{[7N/8]} \), \( S^{[9N/8]} \), \( S^{[15N/16]} \), \( S^{[7N/8]} \), \( S^{[3N/4]} \), \( S^{[N/2]} \), \( S^0 \) by applying the bridge.

A hitting time to the option barrier can be constructed as follows. Given a value \( S^N \) for time \( t_N \) generate a value \( S^{[N/2]} \) \( S^0, S^N \) by applying the bridge. Continue to generate values by binary chop, first to left of the most recently generated value, and then to the right. Whenever a value \( S^i \) is generated, for some time \( t_i \), test if \( S^i \) lies above or below the barrier. If it is below the barrier then no further iteration is required for times greater than \( t_i \); to determine a hitting time only asset values for times less than \( t_i \) need be generated. If it is above the barrier then values both to the left and to the right of \( S^i \) must be generated and tested.

Metwally and Atiya (2002) [37] use a Brownian bridge technique with the Merton jump-diffusion model to simulate sample paths in between jump times. The computational saving in pruning away unnecessary refinements of intervals known not to contain the first hitting time can be considerable. This approach is related to a special case of the contour bridge method when contours are vertical (see section 4).

Neither Dirichlet nor Brownian bridge Monte Carlo work rapidly for exotic barrier options. They sample hitting times only in the fixed discrete set \( \Upsilon \). The contour bridge method permits a more refined value of \( \tau \) to be obtained, with implicitly better sampling of both \( \tau \) and \( \epsilon \), often with significantly faster computation times. This is confirmed by the numerical results in section 5.

4 The contour bridge simulation method

Suppose we are given a set of strictly increasing stopping times \( 0 \equiv \tau^0 < \tau^1 < \ldots \), bounded above by a stopping time \( \tau^\infty \). Set \( \Upsilon = \{ \tau^i \}_{i=0,\ldots,\infty} \). Write \( S_j^i \) for an asset value sampled at the stopping time \( \tau^i \) on the \( j \)th iteration and
\( \tilde{S}_j = \{ S_j \}_{j=0}^{\infty} \) for the jth sample path. A Monte Carlo estimate \( \tilde{c}_t \) for the option value is

\[
\tilde{c}_t = \frac{1}{M} \sum_{j=1}^{M} e^{-\tau(\tilde{t}(S_j) - t)} H \left( \tilde{S}_j \right)
\]

where \( H \left( \tilde{S}_j \right) \) and \( \tilde{t}(S_j) \) are the Monte Carlo approximation to the payoff \( H(\omega) \) and payoff time \( t(\omega) \) computed from \( \tilde{S}_j \). For a suitable chosen set \( \hat{\Upsilon} \), \( \tilde{c}_t \) can be a very good estimator for \( c_t \). We choose \( \hat{\Upsilon} \) to be hitting times to a set of contours.

A contour is the image of a map \( \beta : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \) where \( \beta \) is continuous, \( \beta(0) \in \{0\} \times \mathbb{R}^+ \), and \( \beta \) is 1-1 so that the image does not intersect itself.

In this article we consider contours only of the form \( \beta(t) = (t, \beta'(t)) \). \( \beta \) is determined by \( \beta' \) and we identify \( \beta' \) with \( \beta \). We regard vertical contours, of the form \( \beta(t) = (\alpha) \times \mathbb{R}^+ \) for some \( \alpha \in \mathbb{R}^+ \), as limiting cases of these.

Once the algorithm is initialized, with the construction of a pair of contours that bound the option barrier, intermediate contours are constructed. Hitting times to them are computed by sampling from the contour bridge density. The values of the asset at the hitting times are computed and compared to the values of the option barrier at those times, enabling efficient convergence to a hitting time to the option barrier.

The method generalizes the standard bridge Monte Carlo method in two ways. First, hitting times are not constrained to lie in a finite set \( \hat{\Upsilon} \). This allows values of \( \tau \) to be established with much greater refinement resulting in reduced simulation bias. Secondly, by permitting non-vertical contours very significant variance reductions can be achieved. The method also incorporates a Dirichlet stopping test, enabling the method to benefit from certain advantages of that approach too.

First we discuss the existence of suitable families of contours. Then we describe the general contour bridge method, discussing how it applies to both \( \tau \) - and \( \sigma \)-options. We present two ways of applying it: with general and with single-hit contours, and also present the biggest-bite variant that gives enhanced performance once a pair of contours bracketing a hitting time \( \tau < T \) has been found. Here as elsewhere we describe the contour bridge method as it applies to ‘down’ options, where the initial value of the underlying asset lies above the barrier.

### 4.1 Contours

We are given a model for the evolution of the asset value and suppose that, for this model, there exists an indexed set of contours,

\[
\beta(t \mid \alpha) \in \mathbb{R}^+,
\]

for \( t, \alpha \in \mathbb{R}^+ \), with the following four properties:

1. Ordered. For all \( t, \alpha_1 < \alpha_2 \) implies \( \beta(t \mid \alpha_1) > \beta(t \mid \alpha_2) \).

2. Bounding. There exist contours \( \beta^\infty(t) = \beta(t \mid \alpha^\infty) \) and \( \beta^0(t) = \beta(t \mid \alpha^0) \), with \( S_0 \geq \beta^0(0) \), such that \( \beta^0(t) > b(t) > \beta^\infty(t) \) for all \( t \in [0, T] \).
3. Hittable. Let $\tau^\alpha \equiv \tau (S_0 | \alpha) = \min\{S_t < \beta (t | \alpha) | S_0\}$ be the first passage time of $S$ to the contour $\beta (t | \alpha)$. We suppose that $\tau^\alpha$ can be sampled.

4. Bridgeable. Suppose $\alpha^0 < \alpha_1 < \alpha_2 < \alpha^\infty$, so that $\tau^{\alpha_1} < \tau^{\alpha_2}$. Given $\alpha_1 < \alpha < \alpha_2$ we suppose that it is possible to sample $\tau^{\alpha} | \tau^{\alpha_1}, \tau^{\alpha_2}$, that is, conditional on $\tau^{\alpha_1}$ and $\tau^{\alpha_2}$ it is possible to sample from the bridge distribution $\tau^\alpha$.

It is convenient to require $\Pr [\tau < \infty] = 1$, however it seems possible to relax this requirement in some applications of the method.

The ordered property ensures that hitting times are strictly increasing. The bounding property facilitates the bracketing of barrier hitting times within intervals, and the bridgeable property enables sample paths to built up in non-chronological order, so that sample path refinement can take place only where required.

There are various examples where suitable families of contours can be found.

1. The asset price process $S$ is a geometric Brownian motion and contours are exponential

$$dS_t = rS_t dt + \sigma S_t d\zeta_t, \quad (5)$$

$$\beta (t | \alpha) = \frac{1}{\alpha} \exp (gt), \quad (6)$$

for some fixed growth rate $g$.

In this case the hitting time density is

$$f (t; a, \mu) = \frac{a}{\sigma \sqrt{2\pi t^3}} \exp \left(- \frac{(a - (g - \mu) t)^2}{2\sigma^2 t}\right), \quad (7)$$

where $\mu = r - \frac{1}{2}\sigma^2$ and $a = \ln (S_0 \alpha)$. For $\tau^\alpha \in (\tau^{\alpha_1}, \tau^{\alpha_2})$ the bridge hitting density $f (\tau^\alpha | \tau^{\alpha_1}, \tau^{\alpha_2})$ is

$$f (\tau^\alpha | \tau^{\alpha_1}, \tau^{\alpha_2}) = \frac{1}{\sigma \sqrt{2\pi}} \frac{\alpha_1 \alpha}{\alpha_2} \left(\frac{\tau^{\alpha_1} \tau^\alpha}{\tau^{\alpha_2}}\right)^{-\frac{3}{2}} \exp \left(- \frac{1}{2\sigma^2} \left(\frac{a_1^2}{\tau^{\alpha_1}} + \frac{a_2^2}{\tau^{\alpha_2}} - \frac{a^2}{\tau^\alpha}\right)\right), \quad (8)$$

where $a_1 = \ln \left(\frac{\alpha_1}{\alpha}\right)$, $a_2 = \ln \left(\frac{\alpha_2}{\alpha}\right)$, and $a = \ln \left(\frac{\alpha}{\alpha_1}\right)$. This follows from a related result in Ribeiro and Webber (2004) [45].

2. $S$ is a geometric Brownian motion and contours are horizontal

$$\beta (t | \alpha) \equiv \alpha. \quad (9)$$

This is a limiting case of 1.

3. $S$ is a geometric Brownian motion and contours are vertical. This is the standard Brownian bridge case.

4. $S$ is a geometric Brownian motion with deterministic parameters, $dS_t = r_t S_t dt + \sigma S_t d\zeta_t$, and $\beta (t | \alpha)$ is of the form

$$\beta (t | \alpha) = \frac{1}{\alpha} \exp \left(\int_0^t (r_s - g \sigma_s^2) ds\right) \quad (10)$$

for some parameter $g$. See Brigo, Morini and Tarenghi (2009) [13].
5. $S$ is variance gamma or normal inverse Gaussian and contours are vertical. See Ribeiro and Webber (2003) [44], (2004) [45].

### 4.2 The contour bridge method algorithm

We first describe how the algorithm computes a hitting time $\tau$ to the option barrier, or exits if no $\tau < T$ is found, noting how the algorithm is modified if only $\tau$ is required. For the case of an $\iota$-option requiring the value of $S_T$ we discuss in section 4.2.3 how a value for $S_T | \iota$ can be simulated.

Write $\hat{\tau}$ for a simulated first hitting time of $S$ to the option barrier. The algorithm proceeds as follows:

1. Given a pair of bounding contours, $\beta^0(t) = \beta(t | \alpha^0)$ and $\beta^\infty(t) = \beta(t | \alpha^\infty)$; construct hitting times $\tau^0$ and $\tau^\infty$ to these contours.
   - If $\tau^0 > T$ then the option barrier is not hit and the algorithm stops. If $\tau^\infty < T$ then $\hat{\tau} \in (\tau^0, \tau^\infty)$ and if an $\iota$-option is being valued the algorithm exits.
   - Otherwise set $i = 1$, $\bar{\tau}^1 = \tau^0$, $\underline{\tau}^1 = \tau^\infty$. Apply step 2 to the interval $(\bar{\tau}^1, \underline{\tau}^1)$.

2. At the $i$th step suppose that we have hitting times $\bar{\tau}^i < \underline{\tau}^i$ to bounding contours $\bar{\beta}^i(t) = \beta(t | \bar{\alpha}^i)$ and $\underline{\beta}^i(t) = \beta(t | \underline{\alpha}^i)$, $\bar{\alpha}^i < \bar{\alpha}^i$, with $S(\bar{\tau}^i) = \beta(\bar{\tau}^i | \bar{\alpha}^i) > b(\bar{\tau}^i)$, and we have to determine whether there is a hitting time in the interval $(\bar{\tau}^i, \underline{\tau}^i)$.
   - (a) Test if the algorithm halts. (Stopping rules are discussed in section 4.2.1.)
     - If the algorithm halts and $S(\underline{\tau}^i) = \beta(\underline{\tau}^i | \underline{\alpha}^i) < b(\underline{\tau}^i)$ then the option barrier must be hit in the interval $(\bar{\tau}^i, \underline{\tau}^i)$. Return an approximation $\hat{\tau} = \frac{1}{2}(\bar{\tau}^i + \underline{\tau}^i)$ to the hitting time. Otherwise if $S(\underline{\tau}^i) > b(\underline{\tau}^i)$ return a condition indicating that no hitting time has been found.
     - If the algorithm does not halt then iterate. Set $\alpha^{i+1} = \frac{1}{2}(\bar{\alpha}^i + \underline{\alpha}^i)$ and construct $\tau^{i+1}$ as the hitting time to $\beta(t | \alpha^{i+1})$ conditional on $\bar{\tau}^i$ and $\underline{\tau}^i$.
   - (b) If $\tau^{i+1} > T$ then the option barrier cannot be hit in the interval $(\tau^{i+1}, \underline{\tau}^i)$.
     - Set $\bar{\tau}^{i+1} = \bar{\tau}^i$, $\underline{\tau}^{i+1} = \tau^{i+1}$, increment $i \leftarrow i + 1$ and apply step 2 to the new interval $(\bar{\tau}^i, \underline{\tau}^i)$.
   - (c) If $S(\tau^{i+1}) = \beta(\tau^{i+1} | \alpha^{i+1}) < b(\bar{\tau}^i)$ then the option barrier has been hit in the interval $(\bar{\tau}^i, \tau^{i+1})$. If a bare-$\iota$ option is being valued the algorithm exits immediately.
     - Otherwise set $\bar{\tau}^{i+1} = \bar{\tau}^i$, $\underline{\tau}^{i+1} = \tau^{i+1}$, increment $i \leftarrow i + 1$ and apply step 2 to the new interval $(\bar{\tau}^i, \underline{\tau}^i)$.
   - (d) If $S(\tau^{i+1}) = \beta(\tau^{i+1} | \alpha^{i+1}) > b(\bar{\tau}^i)$ then establish whether the option barrier may be hit in $(\bar{\tau}^i, \tau^{i+1})$ or in $(\tau^{i+1}, \underline{\tau}^i)$.
     - i. If there exists $t \in [\bar{\tau}^i, \tau^{i+1}]$ such that $\beta(t | \alpha^{i+1}) < b(t)$ then it is possible that the option barrier is hit in the interval $(\bar{\tau}^i, \tau^{i+1})$. 

Set \((\check{\tau}^{i+1}, \check{\xi}^{i+1}) = (\hat{\tau}^{i}, \hat{\xi}^{i+1})\), increment \(i \leftarrow i + 1\), and apply step 2 to the new interval \((\check{\tau}^{i}, \check{\xi}^{i+1})\). If this finds a first hitting time \(\hat{\tau} \in (\hat{\tau}^{i}, \hat{\xi}^{i})\) return \(\hat{\tau}\) as the first hitting time in the interval \((\check{\tau}^{i-1}, \check{\xi}^{i-1})\).

ii. If step 2(d)i fails to return a hitting time, then check if it is possible that the option barrier is hit in the interval \((\check{\tau}^{i+1}, \check{\xi}^{i+1})\).

If \(\tau^{i+1} > T\), or if \(\beta(t | \xi^{i}) > b(t)\) for all \(t \geq \tau^{i+1}\), then the option barrier cannot be hit in this interval. Otherwise set \((\check{\tau}^{i+1}, \check{\xi}^{i+1}) = (\hat{\tau}^{i+1}, \check{\xi}^{i})\), increment \(i \leftarrow i + 1\), and apply step 2 to the new interval \((\check{\tau}^{i}, \check{\xi}^{i})\). If this finds a hitting time \(\hat{\tau} \in (\hat{\tau}^{i}, \hat{\xi}^{i})\) return \(\hat{\tau}\) as the first hitting time in the interval \((\check{\tau}^{i-1}, \check{\xi}^{i-1})\).

If step 2(d)ii also fails to return a hitting time then the algorithm returns a value indicating that no hitting time exists in the interval \((\check{\tau}^{i-1}, \check{\xi}^{i-1})\).

4.2.1 The stopping conditions

The algorithm halts if one of two conditions is true. It stops if

1. \(|\hat{\tau}^{i} - \check{\xi}^{i}| < \varepsilon_{\tau}^{1}\), or

2. if \(S(\hat{\tau}^{i})\) and \(S(\check{\xi}^{i})\) both lie above the barrier, then if \(|\hat{\tau}^{i} - \check{\xi}^{i}| < \varepsilon_{\tau}^{2}\), compute (or approximate) \(p = \Pr[\hat{\tau}^{i} < \tau < \check{\xi}^{i} | S(\hat{\tau}^{i}), S(\check{\xi}^{i})]\). Halt if \(p < \varepsilon_{p}\).

\(\varepsilon_{\tau}^{1}\) is the granularity of the hitting times produced by the algorithm. It is the binding condition when a hitting time \(\tau \in (\hat{\tau}^{i}, \check{\xi}^{i})\) has been found. In our numerical work we set \(\varepsilon_{\tau}^{1} = 10^{-10}\). This degree of refinement is possible in a Dirichlet method only if \(T/\varepsilon_{\tau}^{1} = T.10^{10}\) time steps are used, usually an infeasibly large number.

The second condition is introduced to enable the algorithm to end early if the likelihood of hitting the barrier is sufficiently low. Without this condition the algorithm would be inefficient, always taking \(T/\varepsilon_{\tau}^{1}\) steps to establish that the barrier is not hit.

Note that \(p\) is used only as part of a stopping condition. Unlike the Dirichlet method it is not used as part of a bias correction procedure.

Since \(p\) is not usually computable an approximation is used. Suppose that \(\varepsilon_{\tau}^{2}\) is small enough so that over an interval \(\zeta = [t, t + \varepsilon_{\tau}^{2}]\) the option barrier can be approximated reasonably well by a contour, \(b(t) \sim \beta(t | \alpha; g)\) for \(t \in \zeta\). This is a legitimate assumption if, for instance, over \(\zeta\) both the option barrier and a contour are approximately linear. Write \(\tau^{m}\) for the conditional hitting time to the contour, and set \(p^{m} = \Pr[\hat{\tau}^{i} < \tau^{m} < t + \varepsilon_{\tau}^{2} | S(t), S(t + \varepsilon_{\tau}^{2})]\) when \(p^{m}\) can be computed we can approximate \(p\) with \(p^{m}\).

\(p^{m}\) can be computed when \(S\) is a geometric Brownian motion and contours are exponential. Suppose we are given values \(s_{0}\) and \(s_{1}\) of a geometric Brownian motion \(S\) at times \(t_{0}\) and \(t_{1}\), \(t_{0} < t_{1}\), and values \(\beta_{0}\) and \(\beta_{1}\) of an exponential contour \(\beta(t | \alpha)\) also for times \(t_{0}\) and \(t_{1}\). Suppose that \(s_{i} > \beta_{i}, i = 0, 1\). The
conditional hitting probability $p = \Pr \left[ t_0 < \tau < t_1 \mid s_0, s_1, \beta_0, \beta_1 \right]$ of $S$ to $\beta$ is known (see Baldi (1999) [3]). Set $u_i = \ln \left( \frac{s_i}{\beta_i} \right)$ and let

$$g = \frac{1}{t_1 - t_0} \ln \left( \frac{\beta_1}{\beta_0} \right)$$

be the growth rate of $\beta(t \mid \alpha)$. Then

$$p = \exp \left( -\frac{2}{\sigma^2(t_1 - t_0)} u_0 v_{01} \right)$$

where $v_{01} = \ln \left( \frac{s_1}{\beta_1} \right)$.

If $p$ is too large the method would encounter simulation bias. However even with a small value of $p$ the associated stopping condition very significantly reduces execution time. $\epsilon_p^2$ can be set to a value much larger than $\epsilon_p^1$. If the barrier is not hit, in the worse case the method now takes only $T/\epsilon_p^2$ steps before halting.

### 4.2.2 Intersection times

In steps 2(d)i and 2(d)ii it is necessary to know the first and last intersection times $I^L$ and $I^R$ between the method and option barriers in the interval $(\tau^i, \tau^i)$,

$$I^L = \min_{t \in (\tau^i, \tau^i)} \{ \beta(t) = b(t) \}, \quad I^R = \max_{t \in (\tau^i, \tau^i)} \{ \beta(t) = b(t) \}.$$  

A knowledge of $I^L$, $I^R$, for each contour, and whether $\beta(\tau^i)$ and $\beta(\tau^i)$ are above or below $b(\tau^i)$ and $b(\tau^i)$, is sufficient to establish whether iterations are needed to the left or to the right of a hitting time $\tau \in (\tau^i, \tau^i)$. Table 15 illustrates. ‘L’ means that iteration is required on the left, on $(\tau^i, \tau)$, ‘R’ that it is required on the right, on $(\tau, \tau^i)$.

<table>
<thead>
<tr>
<th>Left-hand side conditions</th>
<th>$\tau &gt; I^L$</th>
<th>Right-hand side conditions</th>
<th>$\tau &gt; I^R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta(\tau^i) &gt; b(\tau^i)$</td>
<td>true</td>
<td>$\beta(\tau^i) &gt; b(\tau^i)$</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>L, R</td>
<td>false</td>
<td>L, R</td>
</tr>
</tbody>
</table>

Other speed-ups are possible, depending on the values of $I^L$ and $I^R$ for earlier contours, for instance if $\tau > \tau^{\text{prev}}$, where $\tau^{\text{prev}} = \min_{t \in (\tau^{i-1}, \tau^{i-1})} \{ \beta(t) = b(t) \}$, then refinement is necessary only on $(\tau^i, \tau)$.

Unfortunately it may be expensive to compute $I^L$ and $I^R$. We return to this issue in section 4.3.1.

### 4.2.3 Computing $S_T$

A bare-$\tau$ option does not require a value of the asset price at maturity to be generated. For a general $\tau$-option this is not the case. Indeed the payoff to a general barrier option may depend on both $\tau$ and $S_T$, or indeed on the value
of $S_t$ for some $t \in [0, T]$. It is easy to extend the contour bridge method to simulate values of $S_t$ for some value of $t$ consistent with the values generated for of $\tau^i$. We consider only the case when $t = T$ but it will be clear how the method applies for other times.

First, suppose that $\tau^\infty < T$, so that $\hat{\tau} \in (\tau^0, \tau^\infty)$. Then a value for $S_T$ can be found directly from $S(\tau^\infty) = \beta(S_T | \alpha^\infty)$. If the asset process SDE has a solution the value may be generated in a single step from $\tau^\infty$ to $T$, otherwise a short-step simulation can be used.

Second, suppose that $\tau^0 < T < \tau^\infty$. If the algorithm generates a value of $\tau < T$ then sample $S_T$ from the conditional distribution $S_T | S_r$. Otherwise when the algorithm exits it will have generated a set of hitting times $0 < \cdots < L < T < R < \cdots < \tau^\infty$ (16) where $\tau_L$ is the largest hitting time less than $T$ and $\tau_R$ is the smallest hitting time greater than $T$. Suppose that $\tau_L$ is the hitting time to the contour $\beta(\tau_L | \alpha^L)$ and $\tau_R$ to $\beta(\tau_R | \alpha^R)$.

We need to generate a value $S_T | \tau_L, \tau_R$. If this can be done directly then no further iteration is required. If this is not possible then a numerical approximation can be used. Let $\alpha^M = \frac{1}{2} (\alpha^L + \alpha^R)$ and compute a conditional hitting time $\tau^M$ to $\beta(\tau^M | \alpha^M)$. If $\tau^M > T$ then set $\tau_R = \tau^M$ and repeat the step on the new $(\tau^L, \tau^R)$ pair; if $\tau^M < T$ set $\tau_L = \tau^M$ and again iterate. Exit when $|\tau^L - \tau^R| < \epsilon^4$, returning $S_T = \frac{1}{2} (S(\tau^L) + S(\tau^R))$.

Finally, suppose that $T < \tau^0$. Assuming that one may find a value $\alpha_0$ such that $S_0 = \beta(0 | \alpha_0)$, then the previous method case can be applied to the pair $(\tau^R, \tau^L) = (0, \tau^0)$.

In practice if contours are shallow, with small gradients, the iterative procedure just described may stall. When two shallow contours are close together, it becomes very likely that the process hits an intermediate contour very close to one of the end points, $\tau^0$ or $\tau^1$. The bridge hitting density becomes heavily bimodal with very little likelihood of the next hitting time occurring towards the centre of the interval. The consequence is that as the method evolves the contours get very close together but the values $\tau^0$ and $\tau^1$ may stay far apart and not converge. This is a purely numerical problem. Theoretically the method converges to a true option barrier hitting time $\tau$. In practice, due to rounding errors, the method gets stuck.

The problem particularly effects simulating for $S_T$. One may encounter heavy bias in the outcomes. The problem can be ameliorated by using steep contours, at the cost of increased computation times and reduced efficiency gains. To eliminate bias and alleviate stalling for knock-in options it is possible to first compute the hitting time $\tau$, even though only $i$ is required, and then, when $\tau < T$, simulate $S_T | S_r$. This is unbiased and effective for these options.

4.2.4 Application to a book of options

An advantage of the contour bridge method is that it can be adapted to value simultaneously a book of exotic barrier options, each with different barriers. $\alpha^0$ and $\alpha^\infty$ are chosen to bound the entire set of barriers of options in the book. As a sequence of intermediate values $\alpha^i$ is generated the book-extended method generates refinements in an interval only if there is an option in the book that might hit its barrier in that interval.
There is an additional cost, since refinements need to be found for more intervals, but this is far cheaper than valuing every option individually. Very substantial duplication is avoided, since sample paths are re-used for every option.

4.3 Enhancements to the basic method

The core contour bridge method can be enhanced to increase the degree of variance reduction or to increase its range of applicability. We discuss three variants: the use of single-hit contours, and the biggest-bite and vertical contour variants.

4.3.1 Single-hit contours

In the core method algorithm the most computationally intensive step can be to calculate the first and last intersection times $I_L$ and $I_R$. If no explicit solution to their values is available it is likely to be necessary to conduct an expensive search to find them. Although this search can be accelerated, by tracking previously found first and last intersection times, nevertheless it is a potentially very expensive step.

The method can be simplified if the contours $(t_j)$ can be chosen such that for all $\alpha \in [\alpha^0, \alpha^\infty]$ each contour intersects the option barrier at most once. In this case the conditions in steps 2(d)i and 2(d)ii can be determined from a knowledge of $\beta(\tau^i)$, $\beta(\tau)$ and $\beta(\tau^o)$, and $b(\tau^i)$, $b(\tau)$ and $b(\tau^o)$, without needing to compute $I_L$ and $I_R$.

We refer to this as the single-hit version of the contour bridge method. The single-hit condition may be satisfied only if contours are steep, so that overall there may be reduced efficiency gain. Vertical contours are always single-hit, but are relatively inefficient.

A sufficient condition for single-hit contours to be found is if the minimum slope of each contour is greater than the maximum slope of the option barrier.

For an exponential contour,

$$\beta(t \mid \alpha; g) = \frac{1}{\alpha} e^{\alpha g t},$$

fix $\beta_T \leq b(T)$ and suppose that $\beta_T$ is a maximal value such that a lower bounding contour can be found that takes value $\beta_T$ at time $T$, so that $\beta(T \mid \alpha^\infty; g) = \beta_T$ for some index $\alpha^\infty$. We have $g = \frac{1}{\alpha} \ln (\alpha^\infty \beta_T)$ and the minimum slope is $d_{\min} = \frac{\alpha}{\alpha^\infty}$ at $t = 0$ (for $g > 0$). For all $\alpha \in [\alpha^0, \alpha^\infty]$ this family of contours has minimum slope $d_{\min} = \frac{1}{\alpha\beta_T} \ln (\alpha^\infty \beta_T)$. If the maximum slope of the option barrier in $[0, T]$ is greater than $d_{\min}$ a single-hit contour bridge version is not possible.

4.3.2 The biggest-bite variant

When a sample of the hitting time $\tau$ is needed it is possible to speed-up the single-hit method. Suppose that on an interval $(\bar{\tau}, \underline{\tau})$, with $\underline{\tau} < T$, we have $S(\bar{\tau}) > b(\bar{\tau})$ and $S(\underline{\tau}) < b(\underline{\tau})$ so that $\tau \in (\bar{\tau}, \underline{\tau})$. Call an interval $(\bar{\tau}, \underline{\tau})$ with these properties a bracketing interval. Suppose the current bounding contours are $\beta(t \mid \bar{\alpha})$ and $\beta(t \mid \underline{\alpha})$, $\bar{\alpha} < \underline{\alpha}$. Instead of selecting $\alpha \in (\bar{\alpha}, \underline{\alpha})$ by binary chop, do the following:
1. Set \( \overline{d} = S(\overline{\tau}) - b(\overline{\tau}) \) and \( d = b(\overline{\tau}) - S(\overline{\tau}) \).

2. If \( \overline{d} > d \) choose \( \alpha \) such that \( \beta(\overline{\tau} | \alpha) = b(\overline{\tau}) \). Otherwise choose \( \alpha \) such that \( \beta(\overline{\tau} | \alpha) = b(\overline{\tau}) \).

The biggest-bite method imposes the new selection method once a bracketing interval has been found. At each step the algorithm reduces the range \((\alpha, \omega)\) by the greatest possible amount while preserving the bracketing property.

4.3.3 The vertical contour variant

In this case contours are of the form \( \beta^\alpha \equiv \{ \alpha \} \times \mathbb{R}^+ \) and the methodology applies directly. As there is no overlap between contours one expects this variant to run slowly. On the other hand, using this variant may allow a broader set of models to be used: it requires only the ability to sample \( S_T \) directly from \( S_0 \), and for \( t \in (t_1, t_2) \) the ability to sample \( S_t | S_{t_1}, S_{t_2} \). This variant is an extension of the Brownian bridge method described in section 3.

5 Numerical results

For concreteness and convenience of benchmarking we restrict our examples to the case where the underlying asset \( S = (S_t)_{t \geq 0} \) follows a geometric Brownian motion, but we have noted that the method applies more widely.

We value only a single barrier option at a time, although a significant benefit of the method is its ability to value multiple options simultaneously.

We apply the method to value, first, a benchmark barrier option and, second, a set of exotic barrier options. For the benchmark option, prices can be computable to arbitrary accuracy, much faster than with alternative methods. For the exotic options we also achieve significant variance reduction.

Comparisons are made with a Dirichlet method and with the vertical contour variant. We comment on cases with low and high volatility cases and when \( S_0 \) is close to or far from the barrier.

All options are ‘down’-style, so that \( S_0 > b(0) \). The final maturity time for every option is \( T = 1 \) year. Parameter values for the asset process are \( S_0 = 85 \), \( r = 0.05 \), and \( \sigma = 0.2 \).

5.1 The benchmark option

We value a benchmark barrier option to confirm the accuracy of the new method, and to gauge its performance compared with existing alternative methods. The option we value is a vanilla first touch rebate, \( \mathcal{O}^R \). This pays a rebate of \( R \) at the hitting time \( \tau = T^b \) to a flat barrier, \( b(t) \equiv b \), conditional on \( \tau \leq T \).

When the underlying asset follows a geometric Brownian motion an explicit solution for \( \mathcal{O}^R \) is available (Rubinstein (1992) [49], Lerche (1986) [30]). Set

\[
z = \frac{1}{\sigma \sqrt{T}} \ln \left( \frac{b}{S_0} \right) + \delta \sigma \sqrt{T},
\]

then

\[
\mathcal{O}^R (S_0) = R \left[ \left( \frac{b}{S_0} \right)^{\psi + \delta} N(z) + \left( \frac{b}{S_0} \right)^{-\psi - \delta} N \left( z - 2\delta \sigma \sqrt{T} \right) \right]
\]

14
where
\[
\psi = \frac{1}{\sigma^2} \left( r - \frac{1}{2} \sigma^2 \right),
\]
\[
\delta = \sqrt{\psi^2 + \frac{2r}{\sigma^2}}.
\]

Results are presented in table 1. The option has barrier level \( b = 80 \), and rebate \( R = 5 \), maturing at \( T = 1 \). The asset process has \( S_0 \in \{80.5, 85, 100, 115\} \) with \( r = 0.05 \), with \( \sigma = 0.2 \). The values of the explicit solutions are displayed beneath the asset value.

Four methods are compared: the standard Dirichlet method described in section 3; the contour bridge method with vertical contours; the single-hit contour bridge method; and the biggest-bite variant. Results are given for the single-hit method and the biggest-bite variant for four different values of \( g \), these are chosen so that \( \beta(T \mid \alpha^\infty; g) = 80 \) in every case and \( \beta(0 \mid \alpha^\infty; g) = 75, 50, 20, 1 \), respectively. Contours are steeper the greater the value of \( g \).

Each value is computed using \( M = 10^6 \) sample paths. The contour methods use tolerances \( \varepsilon_1^c = 10^{-10}, \varepsilon_2^c = 10^{-10} \), and \( \varepsilon_p = 10^{-10} \). The standard Dirichlet method uses \( N = 1000 \) time steps.

The entries in the table show the option value produced by each method, the standard error in round brackets, the computation time in square brackets, the bias in curly brackets, and the efficiency gain in bold. Values other than the benchmark option values are rounded to two significant figures.

The bias \( b = \frac{\hat{c} - c}{\sigma} \) of a method is the number of multiples of its standard error, \( \sigma \), a simulated option value, \( \hat{c} \), is away from the true option value, \( c \). Values of \( b \) in the range \( \pm 2 \) denote no bias (to a reasonable level of confidence). Consistently large absolute values of \( b \) indicate the presence of bias.

The efficiency gain of method A over method B is \( e^{AB} \),
\[
e^{AB} = \frac{(se_B)^2 t_B}{(se_A)^2 t_A}, \tag{19}
\]
where \( se_A \) and \( se_B \) are the standard errors of method A and method B, produced in times \( t_A \) and \( t_B \). \( e^{AB} \) is the multiple of the time taken by method A for method B to achieve the same standard error. Efficiency gains in the tables are with respect to the standard Dirichlet method, shown with an efficiency gain of 1. Gains of less than 1 indicates that the method is less efficient than the Dirichlet method.

The results in the tables show no evidence of bias. Standard errors are constant across all methods. The vertical contour method performs poorly in all cases. Even at its best, when \( S_0 \) is far from the barrier, it is only around two thirds the efficiency of the standard Dirichlet method. Except when \( S_0 \) is close to the barrier the contour methods always produce efficiency gains, often large.

When \( S_0 \) is close to the barrier the single-hit method does not perform well. The Dirichlet method, which exits early if the barrier is hit, has an in-built advantage when only a single option is being valued. If a book were being valued, so that the Dirichlet method would usually need to generate an entire sample path, the contour bridge method would display larger efficiency gains.
<table>
<thead>
<tr>
<th>$S_0$, exact.</th>
<th>Standard Dirichlet</th>
<th>Vertical contour variant</th>
<th>Single-hit contour bridge method</th>
<th>Biggest-bite variant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$g$</td>
<td></td>
<td></td>
<td>$g$</td>
</tr>
<tr>
<td>80.5</td>
<td>4.8460 (0.0008)</td>
<td>4.8462 (0.0007)</td>
<td>4.8470 (0.0009)</td>
<td>4.8458 (0.0009)</td>
</tr>
<tr>
<td>4.8458</td>
<td>(0.0008)</td>
<td>(0.0009)</td>
<td>(0.0008)</td>
<td>(0.0008)</td>
</tr>
<tr>
<td>[33]</td>
<td>[280]</td>
<td>[35]</td>
<td>[39]</td>
<td>[48]</td>
</tr>
<tr>
<td>{0.26}</td>
<td>{0.41}</td>
<td>{1.4}</td>
<td>{0.8}</td>
<td>{0.26}</td>
</tr>
<tr>
<td>1</td>
<td>0.12</td>
<td>0.96</td>
<td>0.70</td>
<td>0.50</td>
</tr>
<tr>
<td>85</td>
<td>3.5934 (0.0022)</td>
<td>3.5912 (0.0022)</td>
<td>3.5949 (0.0022)</td>
<td>3.5943 (0.0022)</td>
</tr>
<tr>
<td>3.5932</td>
<td>(0.0022)</td>
<td>(0.0022)</td>
<td>(0.0022)</td>
<td>(0.0022)</td>
</tr>
<tr>
<td>[210]</td>
<td>[510]</td>
<td>[30]</td>
<td>[43]</td>
<td>[69]</td>
</tr>
<tr>
<td>{0.11}</td>
<td>(0.95)</td>
<td>(-0.89)</td>
<td>(-0.79)</td>
<td>(-0.69)</td>
</tr>
<tr>
<td>1</td>
<td>0.41</td>
<td>6.9</td>
<td>4.8</td>
<td>3.0</td>
</tr>
<tr>
<td>100</td>
<td>1.0802 (0.0020)</td>
<td>1.0835 (0.0020)</td>
<td>1.0810 (0.0020)</td>
<td>1.0813 (0.0020)</td>
</tr>
<tr>
<td>1.0812</td>
<td>(0.0020)</td>
<td>(0.0020)</td>
<td>(0.0020)</td>
<td>(0.0020)</td>
</tr>
<tr>
<td>[430]</td>
<td>[730]</td>
<td>[19]</td>
<td>[30]</td>
<td>[56]</td>
</tr>
<tr>
<td>{-0.48}</td>
<td>[1.1]</td>
<td>[1.2]</td>
<td>[-0.9]</td>
<td>[-1.4]</td>
</tr>
<tr>
<td>1</td>
<td>0.59</td>
<td>22</td>
<td>14</td>
<td>7.60</td>
</tr>
<tr>
<td>115</td>
<td>0.2531 (0.0011)</td>
<td>0.2546 (0.0011)</td>
<td>0.2551 (0.0011)</td>
<td>0.2546 (0.0011)</td>
</tr>
<tr>
<td>0.2537</td>
<td>(0.0011)</td>
<td>(0.0011)</td>
<td>(0.0011)</td>
<td>(0.0011)</td>
</tr>
<tr>
<td>[460]</td>
<td>[750]</td>
<td>[12]</td>
<td>[20]</td>
<td>[37]</td>
</tr>
<tr>
<td>{-0.54}</td>
<td>[0.84]</td>
<td>[0.55]</td>
<td>[1.3]</td>
<td>[0.87]</td>
</tr>
<tr>
<td>1</td>
<td>0.62</td>
<td>40</td>
<td>24</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 1: Benchmark valuation, rebate, flat barrier.
The contour bridge method exhibits increasing efficiency gains as the contours become shallower. Efficiency gains increase as the initial asset moves further away from the barrier, to a maximum of 360. Gains from the biggest-bite variant are usually significantly larger than those from the single-hit method. Other experiments verify that efficiency gains tend to decrease slightly as volatility increases, but they remain substantial.

5.2 The exotic options

We value two types of options with four types of non-constant barriers. The options are:

1. A first touch rebate, $O^R$. This pays a rebate of $R$ at the hitting time $\tau \equiv \tau^b$ to the barrier $b(t)$, conditional on $\tau \leq T$. 

2. A knock-in put option, $O^{IP}$. The payoff $H_T$ at the maturity time $T$ is $H_T = (X - S_T)^+ 1_{\tau \leq T}$ for some strike $X$.

The barriers are:

1. A linear barrier, $b_{lin}^L$,
   \[ b_{lin}^L = l + st. \]  
   Two linear barriers are considered: an increasing linear barrier with $l = 80$, $s = 10$, and a decreasing linear barrier with $l = 80$, $s = -10$.

2. A bull spread barrier, $b_{bs}^\alpha$,
   \[ b_{bs}^\alpha = l + (u - l) N(w(t - c)), \quad u > l \]  
   where $N(.)$ is the standard normal distribution function, with $l = 80$, $u = 90$, $w = 8$ and $c = 0.4$.

3. A quadratic concave barrier, $b_c^\beta$,
   \[ b_c^\beta = a + bt + ct^2, \]  
   with $a = 80$, $b = 30$ and $c = -30$.

4. An linear sine barrier, $b_{ls}^\gamma$,
   \[ b_{ls}^\gamma = l + st + a\sin(b(t + c)), \]  
   Two cases are considered: an increasing linear sine barrier with $l = 79$, $s = 10$, $a = 5$, $b = 15$, $c = 0.2$, and a decreasing linear sine barrier with $s = -10$.

Five of these six barriers are plotted in figure 2, together with bounding pairs of exponential contours. In the figure $\alpha_0 = (\alpha^0)^{-1}$, $\alpha_\infty = (\alpha^\infty)^{-1}$.

The sine-barriers, with their cyclical variation, could find natural applications in commodity markets where asset values can also exhibit considerable seasonality.
Table 2: Non-constant barriers
5.3 Numerical results

In the tables that follow, benchmark values were obtained using a standard Dirichlet Monte Carlo method with $10^8$ sample paths and up to $10^2$ time steps. Computation times were up to 15 hours for each value. In the majority of cases the quoted values are unbiased within standard error. In the decreasing linear sine case, however, the result may still be biased by an amount greater than the standard error.

The contour bridge method uses $\varepsilon_1 = 10^{-10}$, $\varepsilon_2 = 10^{-4}$ and $\varepsilon_p = 10^{-10}$. All (non-benchmark) results are obtained with $M = 10^6$ sample paths; the Dirichlet method uses $N = 512$ times steps.

Tables 3 and 4 present results for the rebate $\mathcal{O}^R$ and the knock-in put, $\mathcal{O}^{IP}$, for the six option barriers described in section 5.2. The tables show results for the single-hit contour bridge method, the biggest-bite variant, and the standard Dirichlet method. The vertical contour variant is significantly slower than the Dirichlet method, achieving gains of only around $1/4$; results for it are not shown. The results in the tables show no evidence on bias.

Values used for the contour growth rate $g$ are the smallest values for which the single-hit contour bridge method is possible. $\alpha^0$ is chosen so that $\beta \left( 0 \mid \alpha^0; g \right) = S_0$; $\alpha^\infty$ is chosen so that $\beta \left( 0 \mid \alpha^\infty; g \right) = b(T)$.

The single-hit method achieves good results with the linear barrier. Although the efficiency gains obtained with the more difficult barriers are only around 2 or 3, nevertheless they are worthwhile. The biggest-bite variant achieves gains 2 to 3 times greater than the single-hit method.

The knock-in option values in table 4 were produced by simulating values of $S_T$ directly from those of $S_r$, when these are available, as described in section 4.2.3. If instead the iterative method is always used biases are introduced.
Other tests were performed to explore efficiency gains when $S_0$ is either close to or far from the barrier value $b(0)$.

All methods have difficulty when initial values are close to the barrier. The biggest-bite method marginally out-performs the Dirichlet method but the circumstances of the comparison endow an advantage to the Dirichlet method. When the initial asset value is close to the barrier, or the barrier rises sharply during the life of the option, the probability of hitting the barrier is high. Since the implementation of the Dirichlet method used in this article exits early when the barrier is hit its computation time decreases markedly in this case. A Dirichlet method forced always to compute an entire sample path, as it would have to if applied to value a book of options, would have considerably larger run times. The efficiency gain to the contour bridge method would then be greater.

Gains to the contour bridge method increase considerably as $S_0$ moves away from $b(0)$. Gains for the biggest-bite method are very respectable, around 3 to 4 times greater than the single-hit method. Even for the worse cases, the sine-barriers, variance reductions into double figures are obtained.

### 6 Conclusion

We have described a novel simulation method, the contour bridge method, that we have applied to the valuation of barrier options with exotic barriers. The results in section 5 demonstrate that the contour bridge method generates efficiency gains over the Dirichlet method, often considerable. We have argued that the contour bridge method yields significantly greater efficiency gains when applied to a book of options. The method can also be extended to apply to discrete and partial barrier options.
The method is unbiased when applied to rebate and knock-in style options. Knock-out options pose difficulties for the method if it not possible to sample directly from a certain conditional distribution.

The biggest-bite version of the method can produce very large efficiency gains. The greatest efficiency gains are available when volatility is lower, but gains are still good when volatility is high. The form of the option barrier affects the gains achievable. The best gains are made with linear barriers, but reasonable gains are also made with the much more difficult linear-sine barriers.

Our examples have focused on assets following a geometric Brownian motion but the method applies more widely to situations where contours with the required sampling properties can be found. These include assets with variance gamma or NIG process.

The method relies upon the existence of contours with certain amenable mathematical properties. This paper has investigated only single barrier options but the method would extend to value double barrier options when suitable contours are available.

We conclude that the contour bridge method is a valuable addition to the set of simulation methods that can be applied to value barrier options, enabling exotic barrier options to be priced with much greater confidence.
References


