Valuing Continuous Barrier Options on a Lattice solution for a Stochastic Dirichlet Problem*

Grace Kuan
University of Exeter
School of Business and Economics
Streatham Court, Rennes Drive
Exeter, EX4 4PU
Tel: +44 1392 263203
Email: G.C.Kuan@exeter.ac.uk

Nick Webber
City University
Cass Business School
London EC2Y 8HB
United Kingdom
Tel: +44 20 7040 5171
Email: Nick.Webber@city.ac.uk

June 26, 2003

First version: January 2003

Abstract

The stochastic Dirichlet problem computes values within a domain of certain functions with known values at the boundary of the domain. When applied to valuing barrier options, solutions are expressed as expected discounted payoffs achieved at hitting times to the boundary of the domain.

We construct a lattice solution to the stochastic Dirichlet problem. In between time steps on the lattice, the lattice process is assumed to have the bridge distribution of the underlying stochastic process.

We apply the Dirichlet lattice to valuing barrier options. A plain simple scheme converges very slowly. We find that the Dirichlet lattice is considerably faster than a plain lattice scheme, converging to 2 decimal places in only several hundred time steps.

The Dirichlet lattice can directly value knock-in barrier options, including knock-in Bermudan barrier options which cannot normally be priced by a plain lattice method. It values Bermudan barrier options and barrier options with non-linear barriers equally quickly. We present results demonstrating the superiority of the Dirichlet lattice over both a plain lattice method and a conditional Monte Carlo method.

*We wish to thank participants at a City University seminar for comments on an early version of this paper. The paper has also benefited from conversations with Gordon Gemmill and Paul Dawson.

1 Introduction

We describe a lattice method that computes knock-in and knock-out continuous single and double barrier option values by a exploiting a knowledge of hitting time distributions to the barrier. The method can be used with non-constant barriers and with knock-in and -out Bermudan barrier options. It is considerably faster than ordinary lattice formulations. Since the lattice is a discrete approximation to a solution of a stochastic Dirichlet problem, we call it a Dirichlet lattice.

The Dirichlet lattice process approximates a continuous process. It takes a discrete set of values at each time step, but at intermediate times is distributed according to the bridge process of the underlying continuous process.

Barrier options are widely traded in foreign exchange markets. They can be bought to allow a view to be taken on the future direction of an FX rate, or as a cheaper alternative to a standard option. Vanilla barrier options with a fixed continuous barrier where the underlying asset has a geometric Brownian motion can be valued using formulae developed by Merton (73) [12] and Reiner and Rubinstein (91) [14].

More complex barrier instruments, perhaps with early exercise provision or more complex payoffs, usually need to be valued by a numerical method. Both Monte Carlo and lattices methods can be used.

A simple Monte Carlo method can suffer from simulation bias, caused by the possibility of the barrier being hit in between time steps. Simulation bias in a Monte Carlo method can be reduced using the idea of Beaglehole, Dybvig and Zhou (97) [4] and El Babsiri and Noel (98) [3]. They showed how a knowledge of the extremes distribution of a Brownian bridge could be used to correct for simulation bias.

Ordinary lattice methods can suffer from a similar problem to plain Monte Carlo, discretization bias, that leads to slow convergence. This arises because nodes on the lattice do not exactly correspond to the barrier level. As the number of time steps increases the lattice will converge to the continuous time value. However, while this is happening, the levels of nodes on the lattice will shift relative to the barrier, leading to slow, non-uniform and unpredictable convergence properties.

Several authors have suggested methods to overcome discretization bias in lattice methods to value barrier options. These may involve altering the location of nodes on the lattice (for instance Boyle and Lau (94) [5], Ritchken (95) [16], Cheuk and Vorst (96) [7].)

Figlewski and Gao (99) [8] described the adaptive mesh method, a lattice with refined branching near the barrier. This method has significantly improved convergence. Ahn, Figlewski and Gao (99) [1] applied it to value discrete barrier options. As it is refined the adaptive mesh is approximating with increasing accuracy the conditional hitting distribution. The Dirichlet lattice exploits a direct knowledge of the hitting distribution.

Broadie, Glasserman and Kou (97) [6] show how the value of a continuous barrier option is related to values of discrete barrier options with shifted barrier levels. This allows the value of a continuous barrier option to be found by extrapolation from the more easily computed values of discrete barrier options. While this elegant method, extended by Horfelt (03) [10], works well with vanilla barrier options it does not readily generalize to more complex barrier options, for instance those with non-constant or partial barriers. We apply the Dirichlet lattice to options with time varying barriers and Bermudan barrier options, achieving very good results compared to benchmark methods.

Andricopoulos, Widdicks, Duck and Newton (03) [2] use a quadrature method of numerical integration to value options by backwards induction, reporting excellent results with, for instance, discrete barrier options. The method seems particularly appropriate for options with discrete reset times, but it appears to be significantly slower for continuously time varying barrier or continuous exercise conditions. No applications are given to continuous barrier options in the paper. Furthermore, since the method critically depends on an optimal placement of nodes (to avoid problems with vanishing derivatives), it appears that the method must be applied 'case by case' for every strike and maturity variation, limiting its application in practice. By contrast, the Dirichlet lattice needs no adjustment and can use both forward and backwards induction to value 'in' and 'out' type barrier options, enabling it to value many options simultaneously. The Dirichlet lattice can value vanilla barrier options with a single time step, extending the quadrature method to this case.

The next section describes how the Dirichlet lattice can be constructed. We show how a Brownian bridge hitting times distribution can be exploited to construct lattices with considerably reduced discretization bias. Section three presents numerical results. We benchmark the lattice to standard knock-in and knock-out barrier options, using both forward and backwards induction. We then apply the lattice to value options with non-linear barriers, and options which knock-in or out on an underlying Bermudan option. We find that the Dirichlet lattice achieves great accuracy with only a few hundred time steps. Section four concludes.

2 The Dirichlet Lattice

The value c_0 of a European derivative security at time t_0 is $c_0 = \mathbb{E}_0 \left[H_T \frac{p_0}{p_T} \right]$ where p_t is the value at time t of a numeraire, H_T is a potentially path dependent payoff function and expectations are taken with respect to the equivalent martingale measure. We suppose interest rates are constant, value r, and use the accumulator account numeraire so that $c_0 = \mathrm{e}^{-rT}\mathbb{E}_0 [H_T]$ under the spot measure. We assume there is a single state variable in the model, an asset value S_t, and we specialize down to the process for S_t following a geometric Brownian motion, $\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t\mathrm{d}z_t$, for a Wiener process z_t.

We consider options that knock-in or knock-out when the asset value hits a domain boundary. For each $t \in [0, T]$ let $D_t = [l_t, u_t] \subset \mathbb{R}$, $l_t < u_t$, be either a closed interval or else a half-closed interval with one of u_t or l_t unbounded. For simplicity, we also suppose that the functions $t \longmapsto u_t$ and $t \longmapsto l_t$ are continuous.

The payoff to a European barrier option takes place at time T, although whether a payoff is made or not depends on whether a barrier value has been breached. We consider the case where rebates are paid or received when the option is knocked in or out, and we also investigate Bermudan barrier options. We only consider options where if a non-zero payoff is made at time T, its value depends only upon S_T and not, for instance, on the hitting time.

2.1 The Stochastic Dirichlet Problem

Given a domain $D \subset \mathbb{R}^N$, a bounded measurable function ϕ on the boundary ∂D, and a stochastic process X_t, the stochastic Dirichlet problem is to find an X-harmonic function $\widetilde{\phi}$ on D such that $\widetilde{\phi}$ agrees with ϕ on ∂D (at least in a limit). When the solution exists it is $\widetilde{\phi}(x) = \mathbb{E}\left[\phi(X_\tau) \mid X_0 = x\right]$ where τ is the hitting time of X_t to ∂D.

Barrier option pricing involves a particular version of the stochastic Dirichlet problem. In general we have a time-varying domain $D_t = [l_t, u_t] \subset \mathbb{R}$ with boundary $\partial_t = \{\{l_t\}, \{u_t\}\}$ and a stochastic process $S_t \in \mathbb{R}$ with $S_0 \in D_0$. Upon hitting the boundary at time τ the barrier option acquires the value $\phi_\tau(S_\tau)$ of some other instrument. $\phi_\tau(S_\tau)$ is zero for a knock-out option. At the option expiry time T, if $\tau > T$, suppose the option has value $\phi_T^{\mathrm{O}}(S_T)$. For European barrier options we seek a C^2 function $\widetilde{\phi}(S, t)$ for $t \leq T$ such that

1. $\widetilde{\phi}(S_\tau, \tau) = \phi_\tau(S_\tau)$, if $\tau \leq T$,
2. $\widetilde{\phi}(S_T, T) = \phi_T^{\mathrm{O}}(S_T)$, if $\tau > T$,
3. $A\widetilde{\phi} = r\widetilde{\phi}$ where A is the generator of S_t.

When D_t and S_t are sufficiently regular the solution is

$$\widetilde{\phi}(S_t, t) = \mathbb{E}_t\left[\exp\left(-\int_t^\tau r_s \mathrm{d}s\right)\phi_\tau(S_\tau)\,\mathrm{I}_{\{\tau \leq T\}} + \exp\left(-\int_t^T r_s \mathrm{d}s\right)\phi_T^{\mathrm{O}}(S_T)\,\mathrm{I}_{\{T < \tau\}}\right]. \tag{1}$$

For a vanilla up-and-in barrier call option the domain $D = (\infty, u]$ is the half open interval with a fixed barrier level u. Upon hitting the barrier the option acquires the value of a European call option. In general the value $\phi_\tau(S_\tau)$ of the option when it is knocked-in may not be known analytically and may need to be computed numerically.

We define unconditional and conditional hitting times to the upper barrier, $\overline{\tau}$, the lower barrier, $\underline{\tau}$, and to both barriers, $\tau = \min\{\overline{\tau}, \underline{\tau}\}$. For $t_1 < t_2$, set

$$\overline{\tau}_{t_1}^{S} = \inf_{s \geq t_1}\{S_s \geq u_s \mid S_{t_1} = S\}, \tag{2}$$

$$\overline{\tau}_{t_1,t_2}^{S,S'} = \inf_{s \geq t_1}\{S_s \geq u_s \mid S_{t_1} = S, S_{t_2} = S'\}, \tag{3}$$

with analogous definitions for $\underline{\tau}$, and set $\tau_{t_1}^{S} = \min\left\{\overline{\tau}_{t_1}^{S}, \underline{\tau}_{t_1}^{S}\right\}$ and $\tau_{t_1,t_2}^{S,S'} = \min\left\{\overline{\tau}_{t_1,t_2}^{S,S'}, \underline{\tau}_{t_1,t_2}^{S,S'}\right\}$.

Let c_0 be the value at time 0 of an option which receives a payoff H at time T of

$$H(S,\overline{\tau},\underline{\tau}) = \overline{H}(S)\,\mathrm{I}_{\left\{\overline{\tau}_0^{S_0}\leq\min\left\{\underline{\tau}_0^{S_0},T\right\}\right\}} + \underline{H}(S)\,\mathrm{I}_{\left\{\underline{\tau}_0^{S_0}\leq\min\left\{\overline{\tau}_0^{S_0},T\right\}\right\}} + H^{\mathrm{O}}(S)\,\mathrm{I}_{\left\{T<\min\left\{\underline{\tau}_0^{S_0},\overline{\tau}_0^{S_0}\right\}\right\}} \tag{4}$$

where $\overline{H}$, $\underline{H}$ and H^{O} depend only on S_T, and I_A is the indicator function. $\overline{H}$ and $\underline{H}$ are payoffs at time T corresponding to ϕ_t and H^{O} corresponds to ϕ_T^{O}. A standard double knock-in call option has $\overline{H} = \underline{H} = (S_T - X)_+$ and $H^{\mathrm{O}} = 0$, and a standard double knock out call option has $\overline{H} = \underline{H} = 0$ and $H^{\mathrm{O}} = (S_T - X)_+$. Single barrier options correspond to one or other of u_t and l_t being unbounded.

Let $\overline{c}_0$ be the value at time 0 of the up-and-in option with payoff $\overline{H}\mathrm{I}_{\left\{\overline{\tau}_0^{S_0}\leq\min\left\{\underline{\tau}_0^{S_0},T\right\}\right\}}$, $\underline{c}_0$ be the value of the down-and-in option with payoff $\underline{H}\mathrm{I}_{\left\{\underline{\tau}_0^{S_0}\leq\min\left\{\overline{\tau}_0^{S_0},T\right\}\right\}}$, and c_0^{O} the value of the knock-out option with payoff $H^{\mathrm{O}}\mathrm{I}_{\left\{T<\min\left\{\underline{\tau}_0^{S_0},\overline{\tau}_0^{S_0}\right\}\right\}}$. Since the three barrier events are mutually exclusive $c_0 = \overline{c}_0 + \underline{c}_0 + c_0^{\mathrm{O}}$. We call $\overline{c}_0$, $\underline{c}_0$ and c_0^{O} the components of the barrier option.

For European style options equation (1) is a direct integration. Let $f_S(S)$ be the density function of S_T and $f_{S,\overline{\tau},\underline{\tau}}(S,\overline{\tau},\underline{\tau})$ be the joint density function of S_T, , $\underline{\tau}_0^{S_0}$ and $\overline{\tau}_0^{S_0}$, both conditional on S_0. Then (1) is

$$c_0 = \mathrm{e}^{-rT}\int_0^T\int_0^T\int_0^\infty H(S,\overline{\tau},\underline{\tau})\, f_{S,\overline{\tau},\underline{\tau}}(S,\overline{\tau},\underline{\tau})\,\mathrm{d}S\mathrm{d}\overline{\tau}\mathrm{d}\underline{\tau}. \tag{5}$$

We now condition on S_T. Write $f_{\overline{\tau},\underline{\tau}}(\overline{\tau},\underline{\tau} \mid S)$ for the conditional density function so that $f_{S,\overline{\tau},\underline{\tau}}(S,\overline{\tau},\underline{\tau}) = f_{\overline{\tau},\underline{\tau}}(\overline{\tau},\underline{\tau} \mid S)\, f_S(S)$ and set

$$\overline{F}_{0,T}^{S_0,S_T}(t) = \Pr\left[\overline{\tau}_0^{S_0} \leq \min\left\{t, \underline{\tau}_0^{S_0}\right\} \mid S_0, S_T\right], \tag{6}$$

$$\underline{F}_{0,T}^{S_0,S_T}(t) = \Pr\left[\underline{\tau}_{t_0}^{S} \leq \min\left\{t, \overline{\tau}_0^{S_0}\right\} \mid S_0, S_T\right], \tag{7}$$

$$F_{0,T}^{S_0,S_T}(t) = \Pr\left[\min\left\{\underline{\tau}_0^{S_0}, \overline{\tau}_0^{S_0}\right\} \leq t \mid S_0, S_T\right], \tag{8}$$

$t \geq 0$, for the conditional distribution functions. $\overline{F}_{0,T}^{S_0,S_T}(T)$ is the probability that u_t is hit before time T and before l_t is hit, conditional on the value of S_T at time T. Then, for instance,

$$\int_0^T\int_0^T \overline{H}(S_T)\,\mathrm{I}_{\left\{\overline{\tau}_0^{S_0}\leq\min\left\{\underline{\tau}_0^{S_0},T\right\}\right\}} f_{\overline{\tau},\underline{\tau}}(\overline{\tau},\underline{\tau} \mid S_T)\,\mathrm{d}\overline{\tau}\mathrm{d}\underline{\tau} \tag{9}$$

$$= \overline{H}(S_T)\int_0^T\int_0^T \mathrm{I}_{\left\{\overline{\tau}_0^{S_0}\leq\min\left\{\underline{\tau}_0^{S_0},T\right\}\right\}} f_{\overline{\tau},\underline{\tau}}(\overline{\tau},\underline{\tau} \mid S_T)\,\mathrm{d}\overline{\tau}\mathrm{d}\underline{\tau} \tag{10}$$

$$= \overline{H}(S_T)\,\overline{F}_{0,T}^{S_0,S_T}(T) \tag{11}$$

so

$$c_0 = \mathrm{e}^{-rT} \int_0^\infty \left(\overline{H}\left(S\right) \overline{F}_{0,T}^{S_0,S}\left(T\right) + \underline{H}\left(S\right) \underline{F}_{0,T}^{S_0,S_T}\left(T\right) + H^{\mathrm{O}}\left(S\right) \left(1 - F_{0,T}^{S_0,S_T}\left(T\right)\right)\right) f_S\left(S\right) \mathrm{d}S \tag{12}$$

When $\overline{F}_{0,T}^{S_0,S}\left(T\right)$ is known, or can be approximated, we can develop a lattice solution to (12), the Dirichlet lattice. We see below that this is possible when S_t has a geometric Brownian motion, for instance.

2.2 Constructing the lattice

We construct a lattice for a Wiener process z_t. Discretise time as $0 = t_0 < \ldots < t_N = T$ where, for simplicity, we assume $\Delta t = t_{j+1} - t_j$, $j = 0, \ldots, N-1$ is a constant. Label nodes at time t_j by the pair (j, i), $i = -N_j, \ldots, N_j$, where $N_j = jK$ for a constant integer K. At time t_j the discretised Wiener process $\widehat{z}_t$ can take values $z_{j,i} \in \{z_0 + i\Delta z\}_{i=-N_j,\ldots,N_j}$ where $\Delta z = \sqrt{\kappa \Delta t}$, for some constant $\kappa > 0$, and $z_0 = 0$. Branching from node (j, i) is to nodes $(j+1, i+k)$, $k = -K, \ldots, K$, with branching probabilities p_k chosen to match the transition density function of the Wiener process. The order of branching is $2K+1$.

When $K = 1$ we have trinomial branching and can choose branching probabilities to be

$$p_k = \begin{cases} \frac{1}{2\kappa}, & k = \pm 1, \\ \frac{\kappa - 1}{\kappa}, & k = 0, \end{cases} \tag{13}$$

setting $\kappa = 3$ to match the first five moments of z_t. In the sequel, except where stated, we assume that branching is trinomial.

We now construct a lattice process $\widehat{S}_t$ for S_t constrained to take discrete values at times t_j but assumed to follow a bridge process at intermediate times. When S_t has a geometric Brownian motion, $\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}z_t$, it has solution $S_t = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)t + \sigma z_t\right)$. On the lattice at node (j, i) the lattice process has value $S_{j,i} = S_{0,0} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)t_j + \sigma z_{j,i}\right)$ where $S_{0,0} = S_0$, the initial value of the asset. Conditional on $S_{j,i}$ the lattice process has the value $S_{j+1,i+k}$ at time t_{j+1} with probability p_k, but at intermediate times $t_j \le t \le t_{j+1}$, the distribution of $\widehat{S}_t$ conditional on $S_{j,i}$ and $S_{j+1,i+k}$ is given by the bridge distribution of S_t. We use a knowledge of this bridge distribution to more accurately model the behaviour of S_t at intermediate times.

We describe option valuation by both forward and backwards induction.

2.2.1 Option valuation by forward induction

If the integrand in (12) is known it may be possible to integrate it directly, or else by a quadrature method (Andricopoulos, Widdicks, Duck and Newton (03) [2]). However, in practical applications (a) the integrand may not be known and (b) an options book can contain many options with different time to maturity and payoff functions, each requiring a separate integration. The forward induction method (Jamshidian (91) [11]) computes values of $\overline{F}_{0,T}^{S_0,S}\left(T\right) f_S\left(S\right)$, *et cetera*, by iterating forward through the lattice, finding these values at each node of the lattice, so that a single pass through the lattice can compute the values of many options simultaneously. This overcomes both problems suffered by direct quadrature methods.

Forward induction cannot be used to value Bermudan options. Backwards induction methods are needed for these cases.

Write $p_{j,i}$ for the probability on the lattice of reaching node (j,i) from the initial node $(0,0)$. $p_{j,i}$ is an approximation to $f_S(S_{j,i})$. Write $\mathcal{B}_{j,i}$ for the set of predecessor nodes to node (j,i),

$$\mathcal{B}_{j,i} = \{l \in \{-N_{j-1},\ldots,N_{j-1}\} \mid (j-1,l) \text{ branches to } (j,i)\}. \tag{14}$$

Then, recursively, $p_{0,0} = 1$ and

$$p_{j,i} = \sum_{l \in \mathcal{B}_{j,i}} p_{j-1,l} p_{i-l} \tag{15}$$

so that $p_{j,i}$ can be constructed at every node (j,i).

Write $\overline{p}_{j,i}$ $(\underline{p}_{j,i})$ for the probability of reaching node (j,i) having been knocked in at the upper (lower) boundary, and $p^{\mathrm{O}}_{j,i}$ for the corresponding probability of reaching node (j,i) without having hit the boundary. Of course $\overline{p}_{j,i} + \underline{p}_{j,i} + p^{\mathrm{O}}_{j,i} = p_{j,i}$. $\overline{p}_{j,i}$, for instance, is an approximation to $\overline{F}^{S_{0,0},S_{j,i}}_{0,t_j}(t_j) f_S(S_{j,i})$. We discuss in a moment how these probabilities can be calculated.

Write $H_{N,i}$, $\overline{H}_{N,i}$, $\underline{H}_{N,i}$ and $H^{\mathrm{O}}_{N,i}$ for the payoffs on the lattice at node (N,i); $H_{N,i} = H(S_{N,i})$, *et cetera*. The forward induction approximation to the integral (12) is

$$c_0 = \mathrm{e}^{-rT} \sum_{i=-N_N}^{N_N} \left(p^{\mathrm{O}}_{N,i} H^{\mathrm{O}}_{N,i} + \overline{p}_{N,i} \overline{H}_{N,i} + \underline{p}_{N,i} \underline{H}_{N,i} \right). \tag{16}$$

As $\Delta t \to 0$ this discrete approximation converges to its continuous time counterpart.

For $t \geq t_j$, set $\overline{F}^{i,l}_{j,k}(t) = \overline{F}^{S_{j,i},S_{k,l}}_{t_j,t_k}(t)$, $\underline{F}^{i,l}_{j,k}(t) = \underline{F}^{S_{j,i},S_{k,l}}_{t_j,t_k}(t)$ and $F^{i,l}_{j,k}(t) = F^{S_{j,i},S_{k,l}}_{t_j,t_k}(t)$. We could try setting

$$\overline{p}_{j,i} = p_{j,i} \overline{F}^{0,i}_{0,j}(t_j), \tag{17}$$

$$\underline{p}_{j,i} = p_{j,i} \underline{F}^{0,i}_{0,j}(t_j), \tag{18}$$

$$p^{\mathrm{O}}_{j,i} = p_{j,i} \left(1 - F^{0,i}_{0,j}(t_j)\right), \tag{19}$$

but there are two problems with these definitions. Firstly, the functions $\overline{F}^{i,l}_{j,k}(t)$, $\underline{F}^{i,l}_{j,k}(t)$ and $F^{i,l}_{j,k}(t)$ may not be known. Secondly, if the lattice is used for valuing several products for different maturities at once (which is feasible since to find $\overline{p}_{N,i}$, say, one computes $\overline{p}_{j,i}$ for all $j \leq N$) arbitrage opportunities may be introduced. This is because the probabilities $\overline{F}^{i,l}_{j,k}(t)$, $\underline{F}^{i,l}_{j,k}(t)$ and $F^{i,l}_{j,k}(t)$

are conditioned only at the initial and final times. At intermediate times t_k, $0 < k < j$, there is no constraint to force the discrete process to take values on the lattice. Setting $p_{j,i|k,l}$ to be the probability on the lattice of reaching node (j, i) conditional on being at node (k, l) at time t_k, we will find, for instance, for $0 < k < j$,

$$p_{j,i}\left(1-\overline{F}_{0,j}^{0,i}(t_j)\right) \neq \sum_{l=-N_k,\ldots,N_k} p_{k,l}\left(1-\overline{F}_{0,k}^{0,l}(t_k)\right) p_{j,i|k,l}\left(1-\overline{F}_{k,j}^{l,i}(t_j)\right). \tag{20}$$

To overcome the this problem, and to facilitate a solution to the first problem, we define distributions $\overline{G}_{0,j}^{0,i}(t)$, $\underline{G}_{0,j}^{0,i}(t)$ and $G_{0,j}^{0,i}(t)$ appropriate for the lattice. Set $\overline{\tau}_{j_1,j_2}^{i_1,i_2} = \overline{\tau}_{t_{j_1},t_{j_2}}^{S_{i_1},S_{i_2}}$ and for $t \geq t_{j_1}$ define

$$\overline{G}_{j_1,j_2}^{i_1,i_2}(t) = \Pr\left[\overline{\tau}_{j_1,j_2}^{i_1,i_2} \leq t \mid S_{t_k} \in \{S_{k,i}\}_{i=-N_k,\ldots,N_k}, j_1 < k < j_2\right], \tag{21}$$

with density $\overline{g}_{j_1,j_2}^{i_1,i_2}(t)$ (also define $\underline{G}$ and G with densities $\overline{g}$ and g). Using $\overline{G}_{0,j}^{0,i}(t)$ and $\underline{G}_{0,j}^{0,i}(t)$ rather than $\overline{F}_{0,k}^{0,l}(t)$ and $\underline{F}_{0,k}^{0,l}(t)$ eliminates the possibility of arbitrage.

We can compute values of $\overline{G}_{j_1,j_2}^{i_1,i_2}(t_j)$, $j \geq j_1$, on the lattice. Clearly, $\overline{G}_{j,j+1}^{l,i}(t_j) = \overline{F}_{j,j+1}^{l,i}(t_j)$ for all j, l, i. Furthermore, since $\min_{t\in[0,T]} |u_t - l_t|$ is bounded away from zero then for a sufficiently small time step Δt the probability of hitting both barriers over a single time step is negligible, so that

$$\overline{G}_{j,j+1}^{l,i}(t_{j+1}) \sim \widetilde{G}_{j,j+1}^{l,i}(t_{j+1}) = \Pr\left[\overline{\tau}_{t_j}^{S} \leq t \mid S_{t_j} = S_{j,i}, S_{t_{j+1}} = S_{j+1,l}\right], \tag{22}$$

which does not depend on $\underline{\tau}_{t_j}^{S}$, and

$$G_{j,j+1}^{l,i}(t_{j+1}) \sim \widetilde{G}_{j,j+1}^{l,i}(t_{j+1}) + \underset{\longleftrightarrow}{G}{}_{j,j+1}^{l,i}(t_{j+1}). \tag{23}$$

$\widetilde{G}_{j,j+1}^{l,i}(t)$ is a significantly more tractable object than $\overline{G}_{j,j+1}^{l,i}(t)$.

We now assume that Δt is sufficiently small so that we can approximate $\overline{G}_{j,j+1}^{l,i}(t_{j+1})$ with $\widetilde{G}_{j,j+1}^{l,i}(t_{j+1})$ and $G_{j,j+1}^{l,i}(t_{j+1})$ with $\widetilde{G}_{j,j+1}^{l,i}(t_{j+1}) + \underset{\longleftrightarrow}{G}{}_{j,j+1}^{l,i}(t_{j+1})$.

To find $\overline{G}_{0,j}^{0,i}(t_j)$, $j > 0$, $i \neq 0$, we first calculate $p_{j,i}$, $\overline{p}_{j,i}$, $\underline{p}_{j,i}$ and $p_{j,i}^{\mathrm{O}}$ forward through the lattice using $\overline{G}_{j,j+1}^{l,i}$.

$\{p_{j+1,i}^{\mathrm{O}}\}_{i=-N_{j+1},\ldots,N_{j+1}}$ can be found immediately from $\{p_{j,i}^{\mathrm{O}}\}_{i=-N_j,\ldots,N_j}$. To get to $(j+1, i)$ without hitting a boundary one must have first reached time t_j without hitting the boundary and then not hit the boundary over the step from time t_j to time t_{j+1}. Hence

$$p_{j+1,i}^{\mathrm{O}} = \sum_{l\in\mathcal{B}_{j+1,i}} p_{j,l}^{\mathrm{O}} p_{i-l}\left(1 - G_{j,j+1}^{l,i}(t_{j+1})\right). \tag{24}$$

Similarly, if one has knocked in by time t_{j+1}, either one knocked in before time t_j or else one knocked in (for the first time) between times t_j and t_{j+1}. Hence

$$\overline{p}_{j+1,i} = \sum_{l \in \mathcal{B}_{j+1,i}} \overline{p}_{j,l} p_{i-l} + \sum_{l \in \mathcal{B}_{j+1,i}} p^{\mathrm{O}}_{j,l} p_{i-l} \overline{G}^{l,i}_{j,j+1} (t_{j+1}), \quad (25)$$

$$\underline{p}_{j+1,i} = \sum_{l \in \mathcal{B}_{j+1,i}} \underline{p}_{j,l} p_{i-l} + \sum_{l \in \mathcal{B}_{j+1,i}} p^{\mathrm{O}}_{j,l} p_{i-l} \underline{G}^{l,i}_{j,j+1} (t_{j+1}). \quad (26)$$

If the barrier condition is not active at time t_0 then $p^{\mathrm{O}}_{0,0} = 1$ and $\overline{p}_{0,0} = \underline{p}_{0,0} = 0$. From this starting point one can now evolve $\overline{p}_{j,i}$, $\underline{p}_{j,i}$ and $p^{\mathrm{O}}_{j,i}$ forward through the lattice up to time t_N, and then use them in (16).

The conditional hitting probabilities are then

$$\overline{G}^{0,i}_{0,j} (t_j) = \frac{\overline{p}_{j,i}}{p_{j,i}}, \; j = 1, \ldots N, \quad (27)$$

$$\underline{G}^{0,i}_{0,j} (t_j) = \frac{\underline{p}_{j,i}}{p_{j,i}}, \; j = 1, \ldots N. \quad (28)$$

Values of $\overline{G}^{i_1,i_2}_{j_1,j_2} (t)$, *et cetera*, for $(j_1, i_1) \neq (0, 0)$ can be computed using conditional probabilities conditioned on starting at node (j_1, i_1) instead of node $(0, 0)$. These values may be needed for certain types of barrier option but we do not investigate these here.

Note that the integral (12) can be computed in a single time step. Set $N = 1$ and let K be large so that $S_{1,k}$, $k = -K, \ldots, K$, is a sample from time T whose range $[S_{1,-K}, S_{1,K}]$ is such that

$$\overline{c}_0 \sim \int_{S_{1,-K}}^{S_{1,K}} \overline{H} (S) \overline{F}^{S_0,S}_{0,T} (T) f_S (S) \, \mathrm{d}S, \quad (29)$$

say, within the accuracy required. If the three functions in the integrand are known and their values computable at $S_{1,k}$, $k = -K, \ldots, K$, the integral can be computed by a numerical integration method, such as Simpson's rule. This is the case when S_t has a geometric Brownian motion. Andricopoulos, Widdicks, Duck and Newton (03) [2] successfully used such a quadrature method, but without the hitting distribution term and not applied to continuous barrier options. Our extension allows us to price vanilla barrier options in a single time step, but our main interest is to price non-standard barrier options for which a single time step is insufficient.

2.2.2 Option valuation by backwards induction

Backwards induction needs to be used if a rebate is paid or payable when the barrier is hit, or if some component of the option can be exercised prior to maturity. A standard lattice method is unable to price Bermudan 'in' type barrier options, but our lattice formulation is able to do so.

We consider American or Bermudan options which knock-in to other American or Bermudan options. Write H_t^{O} for the payoff to the option if it is exercisable at time $t < \tau = \min\{\overline{\tau}, \underline{\tau}\}$. An exercise strategy $\sigma < \tau$ is a stopping time at which exercise takes place. Write $\widetilde{\phi}^{\mathrm{A}}(S_t, t)$ and $\phi_\tau^{\mathrm{A}}(S_\tau)$ for values in the American version of (1),

$$\begin{aligned}\widetilde{\phi}^{\mathrm{A}}(S_t, t) &= \max_\sigma \left\{ \mathbb{E}_t \left[\exp\left(-\int_t^\sigma r_s \mathrm{d}s \right) H_\sigma^{\mathrm{O}} \mathrm{I}_{\{\sigma < \min\{\tau, T\}\}} \right.\right. \\ &\quad + \exp\left(-\int_t^\tau r_s \mathrm{d}s \right) \phi_\tau^{\mathrm{A}}(S_\tau) \mathrm{I}_{\{\tau < \min\{\sigma, T\}\}} \\ &\quad \left.\left. + \exp\left(-\int_t^T r_s \mathrm{d}s \right) H^{\mathrm{O}} \mathrm{I}_{\{T < \min\{\tau, \sigma\}\}} \right] \right\},\end{aligned} \tag{30}$$

where the maximum is taken over all exercise strategies σ (for expositional simplicity we assume that τ and σ do not coincide with each other or with T).

In (30), $\phi_\tau^{\mathrm{A}}(S_\tau)$ is the value acquired by the option upon hitting the barrier, and may itself be the value of an American or Bermudan option with exercise value $\overline{H}_t$ ($\underline{H}_t$) at time $t \geq \tau$ if the upper (lower) barrier was hit.

Over a time step Δt we have

$$\begin{aligned}\widetilde{\phi}^{\mathrm{A}}(S_t, t) &= \max_\sigma \left\{ \mathbb{E}_t \left[\exp\left(-\int_t^\sigma r_s \mathrm{d}s \right) H_\sigma^{\mathrm{O}} \mathrm{I}_{\{\sigma < \min\{\tau, \Delta t\}\}} \right.\right. \\ &\quad + \exp\left(-\int_t^\tau r_s \mathrm{d}s \right) \phi_\tau^{\mathrm{A}}(S_\tau) \mathrm{I}_{\{\tau < \min\{\sigma, \Delta t\}\}} \\ &\quad \left.\left. + \exp\left(-\int_t^{t+\Delta t} r_s \mathrm{d}s \right) \widetilde{\phi}^{\mathrm{A}}(S_{t+\Delta t}, t + \Delta t) \mathrm{I}_{\{\Delta t < \min\{\tau, \sigma\}\}} \right] \right\}\end{aligned} \tag{31}$$

Backwards induction solves (30) by iteration back from time T via an approximation to (31). We suppose that exercise is not possible between times t and $t + \Delta t$. Then

$$\begin{aligned}\widetilde{\phi}^{\mathrm{A}}(S_t, t) &= \max \left\{ H_t^{\mathrm{O}},\ \mathbb{E}_t \left[\exp\left(-\int_t^\tau r_s \mathrm{d}s \right) \phi_\tau^{\mathrm{A}}(S_\tau) \mathrm{I}_{\{\tau \leq \Delta t\}} \right.\right. \\ &\quad \left.\left. + \exp\left(-\int_t^{t+\Delta t} r_s \mathrm{d}s \right) \widetilde{\phi}^{\mathrm{A}}(S_{t+\Delta t}, t + \Delta t) \mathrm{I}_{\{\Delta t < \tau\}} \right] \right\}.\end{aligned} \tag{32}$$

We implement (32) on the lattice.

Write $\underline{H}_{j,i} = \underline{H}_{t_j}(S_{j,i})$, *et cetera*, for payoffs to the option components if exercised at node (j, i) on the lattice. Set $\overline{c}_{N,i} = \overline{H}_{N,i}$, $\underline{c}_{N,i} = \underline{H}_{N,i}$ and

$c_{N,i} = H^{\mathrm{O}}_{N,i}$ and then simultaneously evolve back $\overline{c}_{j,i}$, $\underline{c}_{j,i}$ and $c_{j,i}$. Let

$$\overline{q}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \overline{c}_{j+1,i+k}, \tag{33}$$

$$\underline{q}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \underline{c}_{j+1,i+k}, \tag{34}$$

$$q_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \left(\left(1 - G^{i,i+k}_{j,j+1}\right) c_{j+1,i+k} + \overline{G}^{i,i+k}_{j,j+1} \overline{c}_{j+1,i+k} + \underline{G}^{i,i+k}_{j,j+1} \underline{c}_{j+1,i+k} \right), \tag{35}$$

be the continuation values of the option components. These are option values if the options are not exercised at time t but exercised optimally from time $t+\Delta t$. At exercise times t_j set

$$\overline{c}_{j,i} = \max\left\{\overline{H}_{j,i},\ \overline{q}_{j,i}\right\}, \tag{36}$$

$$\underline{c}_{j,i} = \max\left\{\underline{H}_{j,i},\ \underline{q}_{j,i}\right\}, \tag{37}$$

$$c_{j,i} = \max\left\{H^{\mathrm{O}}_{j,i},\ q_{j,i}\right\}, \tag{38}$$

otherwise $\overline{c}_{j,i} = \overline{q}_{j,i}$, *et cetera*. (36) and (37) give values on the lattice for $\phi^{\mathrm{A}}_t(S_t)$. (35) approximates the expectation in (32). If the option knocks-in at the upper (lower) barrier between times t_j and t_{j+1} the knock-in value at node $(j+1, i+k)$ is the value $\overline{c}_{j+1,i+k}$ ($\underline{c}_{j+1,i+k}$), corresponding to $\phi^{\mathrm{A}}_\tau(S_\tau)$. If it does not knock-in then the option receives the discounted back value of the un-knocked-in value $c_{j+1,i+k}$, which corresponds to $\widetilde{\phi}^{\mathrm{A}}(S_{t+\Delta t}, t+\Delta t)$.

Note that if the values $\phi_\tau(S_\tau)$ or $\phi^{\mathrm{A}}_\tau(S_\tau)$ are known explicitly they can be used directly in (35) and (38) without being evolved back in the lattice.

For a European knock-out option, (35) reduces to

$$c_{j,i} \equiv c^{\mathrm{O}}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \left(1 - G^{i,i+k}_{j,j+1}\right) c^{\mathrm{O}}_{j+1,i+k}. \tag{39}$$

since the contribution to $c_{j,i}$ from node $(j+1, i+k)$ is zero if the option has knocked-out.

For a vanilla European up-and-in call option $G^{i,i+k}_{j,j+1} = \overline{G}^{i,i+k}_{j,j+1}$, and (33) and (35) reduce to

$$\overline{c}^{\mathrm{I}}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \overline{c}_{j+1,i+k}, \tag{40}$$

$$c_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \left(\left(1 - \overline{G}^{i,i+k}_{j,j+1}\right) c_{j+1,i+k} + \overline{G}^{i,i+k}_{j,j+1} \overline{c}_{j+1,i+k} \right), \tag{41}$$

since $c_{j,i}$ is the value of the knock-in option if it has not knocked in by time t_j. If the option knocks-in between times t_j and t_{j+1} the knock in value at node $(j+1, i+k)$ is the vanilla call value $\overline{c}_{j+1,i+k}$.

Note that (35) generalises a result of Reimer and Sandmann (95) [13]. They describe a backwards induction method for knock-in options in which a layer of nodes must lie at the boundary value. If node $(j+1, i+k)$ lies on the boundary, then, effectively, they set $\overline{G}_{j,j+1}^{i,i+k} = 1$, and $\overline{G}_{j,j+1}^{i,i+k} = 0$ otherwise. By contrast the Dirichlet lattice does not require nodes to lie at the boundary level, and exploits a full knowledge of $\overline{G}$.

The expressions above can also be modified if a rebate is paid or earned when the barrier is hit. If a rebate $\overline{R}(t)$ is paid if the upper barrier is hit at time t, and $\underline{R}(t)$ paid at the lower barrier, (35) becomes

$$\begin{aligned} q_{j,i} &= \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} \Big(p_k \left(1 - G_{j,j+1}^{i,i+k}\right) c_{j+1,i+k} \\ &\quad + \overline{G}_{j,j+1}^{i,i+k}(t_{j+1}) \left(\overline{c}_{j+1,i+k} + \overline{R}(t_{j+1})\right) \\ &\quad + \underline{G}_{j,j+1}^{i,i+k}(t_{j+1}) \left(\underline{c}_{j+1,i+k} + \underline{R}(t_{j+1})\right) \Big). \end{aligned} \tag{42}$$

2.3 Interpreting the Dirichlet Lattice

Although the lattice process takes a discrete set of values $\{S_{j,i}\}$, branching to a limited number of successor nodes at each step, by construction the lattice assumes that at intermediate times it follows a bridge process. For particular options, branching probabilities are effectively modified to exploit a knowledge of the bridge process.

Write $p_k^{\mathrm{O},j,i} = p_k \left(1 - G_{j,j+1}^{i,i+k}\right)$. Then both (24) and (39) can be interpreted as branching in a lattice for a knock-out option where the branching probabilities are level dependent, with $p_k^{\mathrm{O},j,i}$ the kth branching probability at node (j,i).

Through an alternative formulation of (40) we can also write down analogous 'probabilities' for knock-in options. Consider a European up-and-in option. Suppose that

$$c_0 = \mathrm{e}^{-rt_j} \sum_{i=-N_j}^{N_j} \overline{p}_{j,i} \widehat{c}_{j,i} \tag{43}$$

for certain values $\widehat{c}_{j,i}$ evolved back on the lattice. From (25)

$$\overline{p}_{j+1,i} = \sum_{l \in \mathcal{B}_{j+1,i}} \overline{p}_{j,l} p_{i-l} \left(1 + \frac{\overline{G}_{j,j+1}^{l,i}(t_{j+1})}{\overline{G}_{0,j}^{0,l}(t_j)} \left(1 - G_{0,j}^{0,l}(t_j)\right)\right) \tag{44}$$

so level dependent branching 'probabilities' are

$$p_k^{\mathrm{I},j,i} = p_k \left(1 + \frac{\overline{G}_{j,j+1}^{i,i+k}(t_{j+1})}{\overline{G}_{0,j}^{0,i}(t_j)} \left(1 - G_{0,j}^{0,i}(t_j)\right)\right) \tag{45}$$

and the evolved-back value $\widehat{c}_{j,i}$ at node (j,i) is

$$\widehat{c}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k^{\mathrm{I},j,i} \widehat{c}_{j+1,i+k}. \tag{46}$$

Note that in (46) the value $\widehat{c}_{j,i}$ is not the value at node (j,i) of an un-knocked-in knock-in option. In fact

$$\widehat{c}_{j,i} = c_{j,i} + \frac{p^{\mathrm{O}}_{j,i}}{\overline{p}_{j,i}} c_{j,i} \tag{47}$$

where $c_{j,i}$ is the knocked-in and $c_{j,i}$ the un-knocked-in option value.

In practice, (46) cannot be used directly near time 0 on the lattice. Consider an up-and-in option and let $j_C = \arg\max_j \{S_{j,j} < u_{t_j}\}$. Up to time j_C every node in the lattice lies beneath the barrier. In a standard lattice, it is not possible for the option to be knocked in (or out) until after time t_{j_C}. In the Dirichlet lattice values of $\overline{G}^{0,i}_{0,j}(t_j)$, $j \le j_C$, can become vanishing small so that $p_k^{\mathrm{I},j,i}$ can become very large, leading to overflow errors. To solve this problem one evolves back only as far as step j_C and then sets

$$c_{0,0} = \mathrm{e}^{-rj_C\Delta t} \sum_{i=-j_C}^{j_C} \overline{p}_{j_C,i} \widehat{c}_{j_C,i}. \tag{48}$$

We do not use this alternative in the sequel.

2.4 Using a terminal correction

The convergence and accuracy of both forward and backwards induction can be improved by using a terminal correction. A terminal correction can be used if for short times to maturity there exists a good approximate analytical solution for the value of the option. For instance, for a time-varying barrier option, at time $T - \Delta t$, the vanilla analytical formula for an option with a constant barrier equal to $u_{T-\Delta t}$ may be a good approximation. One then evolves the lattice only up to time $T - \Delta t$. At each node at time $T - \Delta t$ one assigns an option value equal to the analytical approximation. These values are then evolved back in the lattice, or used in (16).

The affect of the applying a terminal correction is to substitute a (sufficiently) differentiable payoff function for a non-differentiable one, enabling convergence at the theoretically fastest rate (Heston and Zhou (00) [9]).

Note that separate terminal corrections need to be made to each of the component options $\overline{c}_{j,i}$, $\underline{c}_{j,i}$ and $c_{j,i}$.

2.5 Computing Conditional Hitting Probabilities

To be able to construct the lattice we need to be able to approximate the conditional hitting time probabilities $\overline{F}^{S_0,S_T}_{0,T}(T)$, *et cetera.*

$\overline{F}^{w_0,w_T}_{0,T}(T)$ is known when w_t is a Brownian motion and there is a single linear barrier $u_t = a + bt$. We can find $\overline{F}^{w_0,w_T}_{0,T}(T)$ from the distribution of the conditional maximum to the hitting time distribution. Suppose w_t has drift μ and volatility σ. Set $\overline{M}^{w_0,w_T}_{0,T} = \max_{t\in[0,T]}\{w_t \mid w_0, w_T\}$. When both w_0 and

w_T lie beneath the barrier

$$\overline{F}_{0,T}^{w_0,w_T}(T) = \Pr\left[\overline{\tau}_{t_0}^S \leq T \mid u_t = a + bt; w_0, w_T\right] \tag{49}$$

$$= \Pr\left[\overline{\tau}_{t_0}^S \leq T \mid u_t = a; w_0, w_T - bT\right] \tag{50}$$

$$= \Pr\left[\overline{M}_{0,T}^{w_0,w_T-bT} \geq a\right] \tag{51}$$

$$= \exp\left(-\frac{1}{2}\frac{(a-w_0)(a-w_T+bT)}{\sigma^2 T}\right), \tag{52}$$

else $\overline{F}_{0,T}^{w_0,w_T}(T) = 1$.

Beaglehole, Dybvig and Zhou (97) [4] and El Babsiri and Noel (98) [3] apply this result to a Monte Carlo method for valuing barrier options with a constant barrier on an asset following a geometric Brownian motion. The same distribution was also used by Ribeiro and Webber (03) [15] to value barrier and other options when asset returns are driven by Lévy processes.

From (52) the hitting probability of a geometric Brownian motion S_t to a constant barrier u can be found. Set $w_t = \ln\left(\frac{S_t}{S_0}\right)$ and $\widehat{u}_t = \ln\left(\frac{u}{S_t}\right)$. For $u > \max\{S_0, S_T\}$ we have

$$\overline{F}_{0,T}^{S_0,S_T}(T) = \Pr\left[\overline{M}_{0,T}^{S_0,S_T} > u\right] = \Pr\left[\overline{M}_{0,T}^{w_0,w_T} > \widehat{u}_0\right] \tag{53}$$

$$= \exp\left(-2\frac{(\widehat{u}_0 - w_0)(\widehat{u}_0 - w_T)}{T}\right) \tag{54}$$

$$= \exp\left(-2\frac{\widehat{u}_0\widehat{u}_T}{\sigma^2 T}\right), \tag{55}$$

and $\overline{F}_{0,T}^{S_0,S_T}(T) = 1$ when $u \leq \max\{S_0, S_T\}$.

When S_t has a geometric Brownian motion we are able to find a good approximation to the probabilities $\overline{G}_{j,j+1}^{i,i+k}(t_{j+1})$ when the barrier is non-linear. The returns process $R_t = \ln\left(\frac{S_t}{S_0}\right)$ is a Brownian motion whose conditional hitting probabilities to a linear barrier are given by (52). Let $\widehat{u}_t = \ln\left(\frac{u_t}{S_0}\right)$ and $\widehat{l}_t = \ln\left(\frac{l_t}{S_0}\right)$ be barrier levels for R_t. Hitting times of R_t to $\widehat{u}_t$ and $\widehat{l}_t$ are identical to hitting times of S_t to u_t and l_t.

We assume that u_t and l_t are sufficiently regular so that over the time step Δt of the lattice $\widehat{u}_t$ and $\widehat{l}_t$ can be approximated as piece-wise linear functions,

$$\widehat{u}_t \sim a_j + b_j t,\ t \in [t_j, t_{j+1}], \tag{56}$$

$$\widehat{l}_t \sim c_j + d_j t,\ t \in [t_j, t_{j+1}]. \tag{57}$$

We use the hitting probabilities of R_t to these linear barriers, given by (52), as our approximation to the hitting probabilities of S_t to u_t and l_t.

In our numerical applications we use an endpoint approximation, setting $b_j = \frac{\widehat{u}_{t_{j+1}} - \widehat{u}_{t_j}}{\Delta t}$ and $a_j = \widehat{u}_{t_j} - b_j t_j$. This is a continuous approximation with knot value $\widehat{u}_{t_j}$ at time t_j.

3 Numerical Results

In this section we value knock-in and knock-out continuous barrier options on an underlying asset following a geometric Brownian motion. No rebates are earned or paid. The initial asset value is $S_0 = 100$, the riskless interest rate is $r = 0.1$, and the asset volatility is $\sigma = 0.25$.

We benchmark to European knock-in and knock-outs calls maturing in one year, $T = 1$, with a continuous single barrier level of $u = 110, 130$ or 150, with a strike of $X = 100$. There is an explicit solution for the values of these options which enables us to benchmark the method.

We then use the lattice to price Bermudan barrier options and barrier options with time-varying non-linear barriers. We find that the Dirichlet lattice prices more accurately that either a plain lattice method or conditional Monte Carlo.

Lattices for knock-out options are truncated at the barrier level. All lattices, Dirichlet and Plain, are in any case truncated at 8 standard deviations either side of the expected final value of the underlying. The plain lattice method computes values for knock-in barrier options by barrier parity; it computes knock-out values and subtracts them from vanilla option values found from the Black-Scholes formula. Barrier parity does not hold for Bermudan barrier options so we are unable to use the plain method in this case.

In practice $\overline{G}_{j,j+1}^{i,i+k}(t_{j+1})$ is very close to zero if $S_{j,i}$ is some distance below the barrier. If $\overline{G}_{j,j+1}^{i,i+k}(t_{j+1})$ is set to zero when node (j, i) is more than ten layers beneath the barrier there is no difference to machine accuracy in the computed option value, but the computation time is roughly halved. We call this the partial Dirichlet method. This method, with a cut-off ten layers from the barrier, is used to compute the results in this section.

3.1 Benchmark Results

Tables 1 and 2 represent benchmark results for up-and-in and up-and-out barrier call options. A terminal correction is imposed at time $T - 0.01$. The top value in each table entry is the computed option value. Code was written in VBA 6.0 with no special speed-ups. The platform was a 1.8 Ghz pentium 4 PC. For the Dirichlet lattice the top value in square brackets is the time in seconds taken by the backwards induction method, the second value is the time taken by forwards induction.

Forwards induction and backwards induction return identical values. To value a single option forward induction is slightly slower than backwards induction; however, the forwards induction method can value many options simultaneously.

Each table gives comparisons using the conditional Monte Carlo method of Beaglehole, Dybvig and Zhou (97) [4] and El Babsiri and Noel (98) [3] with $100,000$ sample paths. We simulate up to time $T = 1$ in a single time step. The entry in round brackets is the standard error. The conditional Monte Carlo method is far superior the plain Monte Carlo, but it is still relatively inaccurate

Up-and-in calls: Comparison of Methods						
Barrier level:	$u = 110$		$u = 130$		$u = 150$	
Explicit:	14.916		12.692		7.928	
Monte Carlo	14.89 (0.06) [1.2]		12.75 (0.07) [1.2]		7.96 (0.06) [1.2]	
Lattice	Plain	Dirichlet	Plain	Dirichlet	Plain	Dirichlet
$N = 100$	14.886 [0.015]	14.910 [0.031] [0.047]	12.344 [0.016]	12.646 [0.031] [0.047]	7.511 [0.016]	7.896 [0.032] [0.047]
$N = 500$	14.900 [0.13]	14.915 [0.24] [0.45]	12.478 [0.11]	12.697 [0.22] [0.45]	7.715 [0.12]	7.907 [0.22] [0.50]
$N = 1000$	14.902 [0.31]	14.916 [0.64] [1.3]	12.555 [0.31]	12.690 [0.63] [1.3]	7.770 [0.36]	7.920 [0.64] [1.3]
$N = 2000$	14.908 [0.90]	14.915 [1.7] [3.6]	12.590 [0.88]	12.691 [1.8] [4.2]	7.783 [0.95]	7.930 [1.7] [3.6]

Table 1: Benchmark: Knock-in calls

compared to the Dirichlet lattice, despite its ability to use 'long-step' simulation.

For the up-and-in call table 1 shows the Dirichlet lattice is accurate to within 2 decimal places with $N = 1000$ time steps. In about the same amount of time the conditional Monte Carlo method still has a standard error of 0.06 and cannot be considered to be accurate to 1 decimal place. Values for the plain lattice method were computed by in-out parity. The plain lattice results are much inferior to the Dirichlet lattice. For $u = 130$ and $u = 150$ the plain lattice is not correct at $N = 2000$ to even 1 decimal place.

Table 2 compares the Dirichlet lattice method with the plain lattice and with conditional Monte Carlo. For the same number of time steps the plain lattice is a little faster than the Dirichlet lattice, but it is much less accurate. The Dirichlet lattice is accurate to about 2 decimal places at $N = 1000$. The plain lattice method is not yet accurate to 1 decimal place (for $u = 130$ and 150). In about the same computational time the conditional Monte Carlo method still has significant standard error.

Figures 1 and 2 show convergence for the plain and Dirichlet lattices for an up-and-out with $u = 130$ and an up-and-in barrier options with $u = 150$. It is clear that the Dirichlet lattice is very accurate compared to the plain lattice.

3.2 Application to Non-Vanilla Barrier Options

We apply the partial Dirichlet lattice using backwards induction to price Bermudan up-and-in and up-and-out barrier options and barrier options with non-

Up-and-out calls: Comparison of Methods						
Barrier level:	$u = 110$		$u = 130$		$u = 150$	
Explicit:	0.0602		2.284		7.047	
Monte Carlo	0.060 (0.002) [1.2]		2.30 (0.02) [1.2]		7.03 (0.03) [1.2]	
Lattice	Plain	Dirichlet	Plain	Dirichlet	Plain	Dirichlet
$N = 100$	0.0862 [0.016]	0.0657 [0.016] [0.015]	2.596 [0.015]	2.330 [0.016] [0.031]	7.442 [0.016]	7.080 [0.016] [0.015]
$N = 500$	0.0749 [0.13]	0.0611 [0.14] [0.14]	2.492 [0.11]	2.279 [0.14] [0.16]	7.256 [0.13]	7.069 [0.16] [0.16]
$N = 1000$	0.0738 [0.31]	0.0601 [0.34] [0.38]	2.419 [0.33]	2.286 [0.38] [0.41]	7.204 [0.36]	7.055 [0.39] [0.42]
$N = 2000$	0.0677 [0.84]	0.0605 [0.91] [0.99]	2.385 [0.89]	2.284 [1.0] [1.0]	7.192 [0.96]	7.046 [1.1] [1.2]

Table 2: Benchmark: Knock-out calls

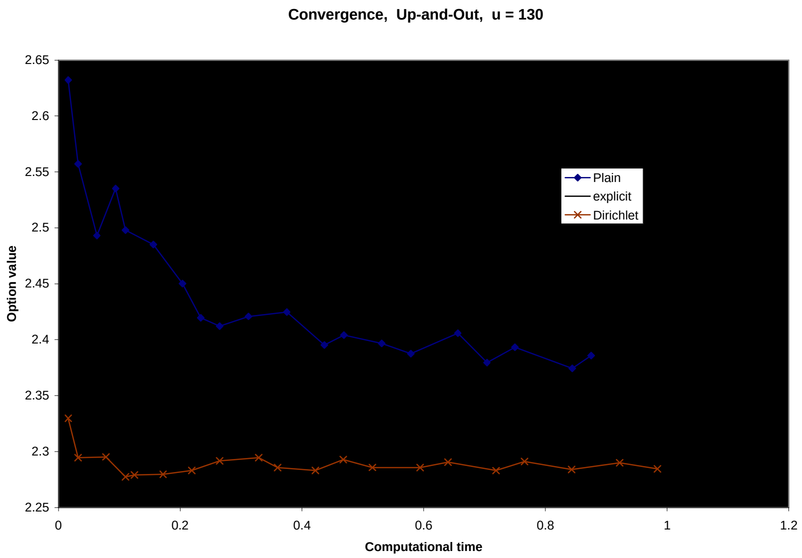

Figure 1: Convergence, plain and Dirichlet lattices, Up-and-Out option.

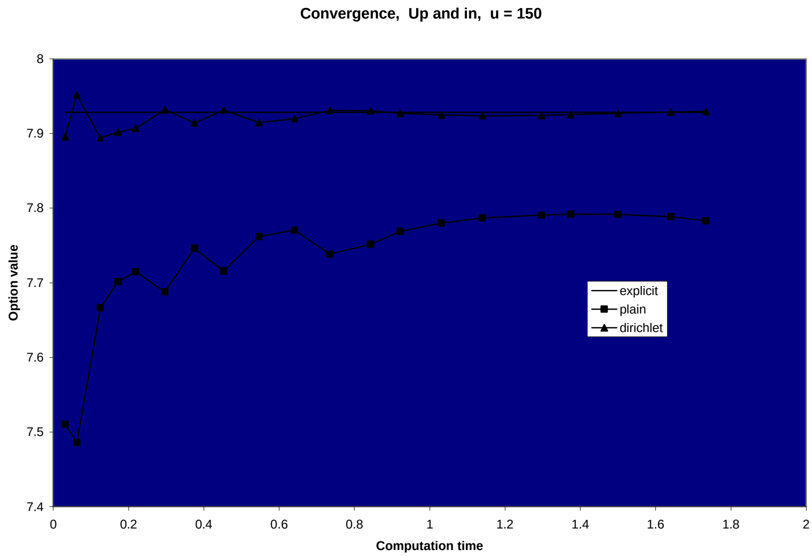

Figure 2: Convergence, plain and Dirichlet lattice, Up-and-In option

constant barriers. The Dirichlet lattice consistently prices more accurately than either conditional Monte Carlo or the plain lattice method.

3.2.1 Bermudan Options

Table 3 shows convergence of the method for Bermudan up-and-out puts with quarterly and monthly resets (Resets $= 4$ and 12). The put has a strike of 100 and one year to maturity. For comparison, values of the corresponding European put ("Explicit" and "Resets $= 1$") are also given. Table 4 gives corresponding knock-in values

It is awkward to use Monte Carlo for Bermudan options so we give a comparison solely with the plain lattice method. A terminal correction is imposed at time $T - 0.01$.

Computation times are not very sensitive to the number of resets and vary relatively little with the barrier level, so they are given with the number of time steps, N. Times shown are for $u = 130$ with 12 resets. The upper entry in each box refers to the Dirichlet lattice, the lower entry to the plain lattice.

At a barrier level of 150 both the Dirichlet and the plain lattice perform similarly. This option knocks out only when it is far out of the money so the advantage of the Dirichlet lattice is less marked. However at lower barrier levels the plain lattice appears to be giving results that are significantly biased. The values found by the Dirichlet lattice appear to be converging, to 2 decimal places, but the plain lattice is not yet accurate to 1 decimal place. The Dirichlet lattice

Up-and-out Bermudan puts									
Barrier:	$u = 110$			$u = 130$			$u = 150$		
Explicit:	3.516			5.350			5.457		
Resets:	1	4	12	1	4	12	1	4	1
$N = 100$									
[0.02]	3.515	4.254	4.380	5.352	6.195	6.340	5.457	6.302	6.4
[0.02]	3.925	4.693	4.823	5.388	6.231	6.377	5.458	6.303	6.4
$N = 500$									
[0.13]	3.533	4.281	4.396	5.350	6.191	6.351	5.457	6.300	6.4
[0.11]	3.715	4.472	4.591	5.368	6.209	6.369	5.457	6.300	6.4
$N = 1000$									
[0.36]	3.526	4.272	4.387	5.350	6.190	6.352	5.457	6.299	6.4
[0.31]	3.663	4.417	4.534	5.364	6.204	6.367	5.457	6.299	6.4
$N = 2000$									
[0.97]	3.519	4.264	4.379	5.350	6.190	6.351	5.457	6.299	6.4
[0.88]	3.631	4.384	4.501	5.360	6.200	6.362	5.457	6.299	6.4

Table 3: Application: Up-and-out Bermudan Puts

takes only slightly longer to run as the plain lattice to run, but it is far more accurate.

The up-and-in Bermudan put values are very small, but nevertheless the Dirichlet lattice appears to converge here also. Since the plain lattice method cannot be used to value up-and-in Bermudan options, table 4 gives no comparison.

3.2.2 Non-Constant Barrier Options

We value two non-constant barrier options. The barriers are:

$$\begin{array}{ll} 1) & u_t = 110 \exp\left(\frac{3}{10}t^2 + \frac{1}{10}t\right), \\ 2) & u_t = 165 - 10 \exp\left(\frac{3}{10}t^2 + \frac{3}{2}t\right). \end{array} \tag{58}$$

Barrier (1) is convex and monotonic increasing from 110 at time 0 to roughly 164 at time 1. Barrier (2) is concave and monotonic decreasing from 155 a time 0 to about 105 at time 1. Payoffs are to vanilla calls with strike $X = 100$. We use partial Dirichlet branching at ten layers from the barrier, pricing by forward induction. The log-barrier is approximated with a piecewise linear endpoint approximation. Monte Carlo is conditional with $100,000$ sample paths and 600 time steps. We do not use a terminal correction in these cases as we do not have an analytical approximation to the option value.

Table 5 gives the results. For most options the Dirichlet lattice is achieving accuracy to 2 decimal places, whereas the plain lattice has plainly not converged. Both plain and Dirichlet lattices have difficulty in valuing the far out of the money knock-out call with barrier (2). With a non-linear barrier, conditional

Up-and-in Bermudan puts									
Barrier level:	$u = 110$			$u = 130$			$u = 150$		
Explicit:	1.946			0.110			0.0028		
Resets	1	4	12	1	4	12	1	4	12
$N = 100$ [0.05]	1.945	2.100	2.155	0.108	0.110	0.112	0.0026	0.0027	0.0027
$N = 500$ [0.5]	1.926	2.078	2.139	0.109	0.112	0.114	0.0028	0.0028	0.0029
$N = 1000$ [1.6]	1.934	2.087	2.148	0.110	0.112	0.115	0.0028	0.0028	0.0029
$N = 2000$ [4.4]	1.940	2.095	2.156	0.110	0.112	0.115	0.0028	0.0028	0.0029

Table 4: Application: Up-and-in Bermudan Puts

Non-Constant Barriers								
Barrier type:	(1)				(2)			
Option type:	Knock-out		Knock-in		Knock-out		Knock-in	
Monte Carlo:	2.90 (0.02) [670]		12.00 (0.06) [670]		0.0486 (0.001) [677]		14.852 (0.06) [676]	
Lattice type:	Dirichlet	Plain	Dirichlet	Plain	Dirichlet	Plain	Dirichlet	Plai
$N = 100$	2.878 [0.02]	3.683 [0.02]	12.108 [0.05]	11.292 [0.02]	0.0804 [0.03]	0.1108 [0.02]	14.905 [0.05]	14.8 [0.0
$N = 500$	2.885 [0.1]	3.278 [0.1]	12.093 [0.4]	11.698 [0.1]	0.0537 [0.1]	0.0632 [0.1]	14.924 [0.4]	14.9 [0.1
$N = 1000$	2.885 [0.4]	3.170 [0.3]	12.089 [1.3]	11.806 [0.3]	0.0497 [0.4]	0.0739 [0.3]	14.924 [1.3]	14.9 [0.3
$N = 2000$	2.888 [1.1]	3.078 [0.9]	12.088 [3.5]	11.897 [0.9]	0.0531 [1.2]	0.0629 [0.9]	14.922 [3.5]	14.9 [0.9

Table 5: Application: Non-constant barriers

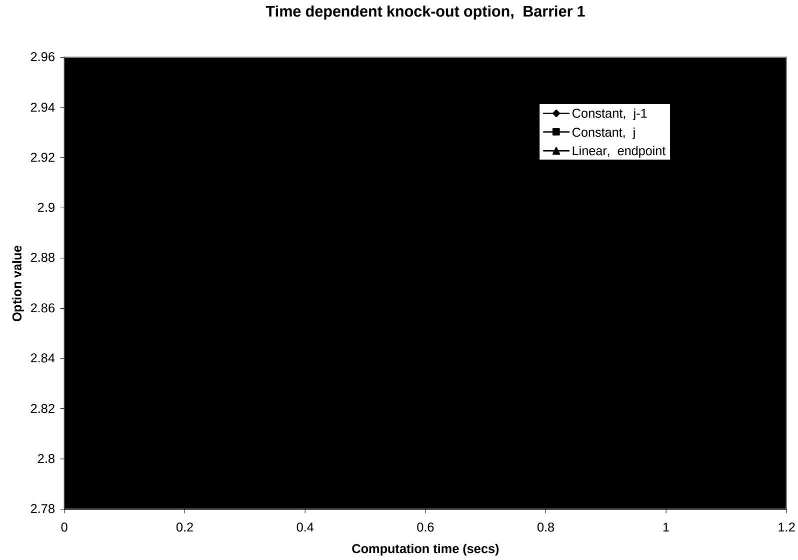

Figure 3: Convergence, Comparison of approximations to a non-linear barrier.

Monte Carlo cannot use a single 'long step' simulation and is painfully slow compared to the lattice.

Figure 3 shows more detailed convergence for the knock-out option with barrier (1). It compares the piecewise linear endpoint approximation with two piecewise constant approximations. Values for the plain lattice are not shown. 'Constant, $j-1$' sets $\widehat{u}_t = \widehat{u}_{t_{j-1}}$, $t \in [t_{j-1}, t_j]$. 'Constant, j' sets $\widehat{u}_t = \widehat{u}_{t_j}$, $t \in [t_{j-1}, t_j]$. For a convex increasing barrier 'Constant, $j-1$' lies beneath the barrier and tends to undervalue up-and-out options. 'Constant, j' lies above it, overvaluing up-and-outs.

Convergence is not uniform, but the piecewise linear endpoint approximation, lying between the piecewise constant approximations, is clearly superior to both.

4 Conclusions

In this paper we have constructed a lattice, based upon a knowledge of conditional hitting time distributions, to value continuous barrier options. We call this a Dirichlet lattice. The Dirichlet lattice benchmarks well to vanilla barrier option values. We applied the lattice to value Bermudan barrier options and barrier options with non-linear barriers. We found that the Dirichlet lattice was considerably more accurate than alternative methods.

Several extensions of the method are possible. By making the stock volatility

$\sigma = \sigma_j$ a deterministic function of the time step, its values can be fixed so that the lattice recovers a term structure of volatility. Similarly if the lattice is implemented for a term structure model, it is possible to calibrate to an observed term structure of interest rates. In the latter case an offset is applied at each time step to recover the market short rate. This translates into a time varying barrier on the lattice.

The Dirichlet lattice is very much superior to a plain lattice method, and to conditional Monte Carlo. The Dirichlet is most advantageous compared to a quadrature method when applied to options with time varying barriers, since in these case a quadrature method requires a large number of time steps.

We find that the Dirichlet lattice in achieving accuracy to 2 decimal places with less than five hundred time steps. It is a simpler alternative to quadrature methods for options with continuous time varying barriers. It can directly value up-and-in Bermudan puts, which is not possible on a plain lattice and difficult with a Monte Carlo method.

References

[1] D-G. Ahn, S. Figlewski, and B. Gao. Pricing Discrete Barrier Options with an Adaptive Mesh. *Journal of Derivatives*, 6(4):33–44, 1999.

[2] A. Andricopoulos, M. Widdicks, P. Duck, and D. Newton. Universal option valuation using quadrature methods. *Journal of Financial Economics*, 67:447–471, 2003.

[3] M. El Babsiri and G. Noel. Simulating path dependent options: A new approach. *Journal of Derivatives*, Winter:65–83, 1998.

[4] D. R. Beaglehole, P. H. Dybvig, and G. Zhou. Going to extremes: Correcting simulation bias in exotic option valuation. *Financial Analysts Journal*, January/February:62–68, 1997.

[5] P. Boyle and S. Lau. Bumping up against the barrier with the binomial method. *Journal of Derivatives*, Summer:6–14, 1994.

[6] M. Broadie, P. Glasserman, and S. Kuo. A Continuity Correction for Discrete Barrier Options. *Mathematical Finance*, 7:325–348, 1997.

[7] T. Cheuk and T. Vorst. Complex barrier options. *Journal of Derivatives*, Fall:8–22, 1996.

[8] S. Figlewski and B. Gao. The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics*, 53:313–351, 1999.

[9] S. Heston and G. Zhou. On the rate of convergence of discrete-time contingent claims. *Mathematical Finance*, 10:53–75, 2000.

[10] P. Horfelt. Extension of the corrected barrier approximation by broadie, glasserman, and kou. *Finance and Stochastics*, 7:231–243, 2003.

[11] F. Jamshidian. Forward induction and construction of yield curve diffusion models. *Journal of Fixed Income*, 1:62–74, 91.

[12] R.C. Merton. Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 73.

[13] M. Reimer and K. Sandmann. A Discrete Time Approach for Auropean and American Barrier Options, journal = Working paper, year = 1995, pages = 1-23,.

[14] E. Reiner and M. Rubinstein. Breaking Down the Barriers. *Risk Magazine*, 4(8), 1991.

[15] C. Ribeiro and N. Webber. Correcting for Simulation Bias in Monte Carlo Methods to Value Exotic Options in Models Driven by Lévy Processes. *Working paper, City University Business School*, 2003.

[16] P. Ritchken. On pricing barrier options. *Journal of Derivatives*, Winter:19–28, 1995.