Valuing Discrete Barrier Options on a Dirichlet Lattice*

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

September 3, 2003

First version: June 2003

Abstract

Discrete barrier options can be valued by quadrature, on a lattice or by Monte Carlo integration. Prices found by an ordinary lattice method will have a large discretisation bias. A good Monte Carlo method will have less bias, but will face difficulties in pricing American style discrete barrier options. Quadrature methods are relatively slow for American barrier options.

We provide a rigorous mathematical framework for valuing discrete barrier options. We show how the Dirichlet lattice of Kuan and Webber can be extended to remove discretisation bias in the lattice valuation of discrete barrier options. Unlike a plain lattice method, the lattice can value American barrier options by backwards or forwards induction and can price a wide range of complex barrier options, including those with multiple and non-constant barrier levels.

Numerical results are given. We conclude the lattice is a relatively simple method of obtaining accurate option values for a wide range of complex European and American discrete barrier options.

*We wish to thank GianLuca Fusai and other participants in a seminar at the University of Piemonte Orientale for helpful comments.

1 Introduction

Discretely reset barrier options are widely traded in the markets. Sometimes analytical solutions are available for their continuously reset counterparts (Reiner and Rubinstein (1991); for double barrier options Kunitomo and Ikeda (1992) give an analytical formula expressed as the sum of infinite series.) but for discretely reset options analytical solutions are rarely available.[1]

Various solution methods have been used for discrete barrier options. These include lattice methods, finite difference methods, Monte Carlo methods, correction methods and quadrature methods. The naive use of these numerical methods can lead to large pricing errors and slow convergence.

Lattice methods are sensitive to the positioning of barrier levels and have non-uniform convergence. To obtain faster convergence, the position of nodes in a lattice relative to the barrier can be adjusted (Ritchken (1995), Cheuk and Vorst (1996), Tian (1999)) but convergence can still be slow and prices may still exhibit considerable bias. Figlewski and Gao (1999) and Ahn, Figlewski and Gao (1999) use refined branching near to the barrier. Their method is able to price significantly more accurately then standard lattices even when barriers are close to the spot price, but may be awkward to implement for complex barrier options.

Finite difference methods have been used by Boyle and Tian (1998) and Zvan *et al.* (2000). These methods can be considered as a form of quadrature method, but since they are precise only in the continuous time limit, they are dominated by quadrature methods when the transition density function of the underlying state variable is known or can be adequately approximated.

Monte Carlo methods are considered by, for instance, Andersen and Brotherton-Ratcliffe (1996), Beaglehole *et al.* (1997) and Baldi *et al.* (1999). Care needs to be taken in these cases to correct for simulation bias. Even so, Monte Carlo methods can be slow, since so-called "long-step" methods - evolving to the final time in a single time step - cannot be used.

Quadrature methods, such as those of Ait-Sahalia and Lai (1997), (1998), Sullivan (2000), Andricopoulos *et al.* (2003), and Fusai and Recchioni (2003), solve for option prices by expanding out the option value as a series of nested integrals. Each integral is then computed by numerical integration. Although reasonably effective for European style options, quadrature methods are slow when used with American style options.

The analytical correction method of Broadie *et al.* (1997), (1999), elaborated by Kou (2001) and Hörfelt (2003), relates the values of discrete barrier option values to the values of the corresponding continuously reset barrier option. (See also Heynen and Kat (1994)). An alternative method is due to Wei (1998). Broadie *et al.* (1999) describe a trinomial lattice for discrete barrier options using the correction and a shift of nodes. The performance of these methods varies depending upon the position of the barrier and the frequency of resets.

[1] Formulae involving multi-dimensional distribution functions usually require numerical solution.

Other methods include the Markov Chain method of Duan *et al.* (2003). This combines features of a Monte Carlo method, a quadrature method, and a lattice method.

In this paper we apply the Dirichlet lattice of Kuan and Webber (2003) to value discretely reset barrier options. Kuan and Webber describe the use of the Dirichlet lattice to value continuous barrier options. The Dirichlet lattice approximates an underlying continuous time process by one taking discrete values at a discrete set of times. At intermediate times the lattice variable has a distribution given by a bridge distribution.

The advantage of the Dirichlet lattice is its simplicity, its relative lack of bias, and its ability to price a very wide range of generalised barrier options. It is faster than Monte Carlo and less complex than finite difference or quadrature methods. It does not require special positioning of nodes, and it can value American barrier options as easily as European options.

The next section describes the construction of the Dirichlet lattice. We show how a Dirichlet lattice can be used to value single, double, and more complex discrete barrier options with considerably reduced discretization bias. Section three presents numerical results. We value discrete knock-in and knock-out barrier options, using both forward and backwards induction. The method benchmarks accurately to values found by previous authors, in particular to Broadie *et al.* (1999) and to Fusai and Recchioni (2003). We then apply the lattice to value Bermudan and other complex barrier options. We find that the discrete lattice achieves great accuracy with a very small number of time steps. Section four concludes.

2 Discrete Barrier Options and the Dirichlet Lattice

We assume there is a single state variable $S = (S_t)_{t\geq 0}$ representing an asset value. We suppose interest rates are constant, value r, and that S follows a geometric Brownian motion, $\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}z_t$, for a standard Brownian motion $z = (z_t)_{t\geq 0}$ under the martingale measure associated with the accumulator account numeraire. The state space Ω is equipped with the completed canonical filtration $\mathcal{F} = \{\mathcal{F}_t\}$ induced S. We will identify $\omega \in \Omega$ with paths of S.

Let $0 = t_0 < T_1 < \ldots < T_Q \leq T$ be a set of reset dates and let c_0 be the value at time t_0 of a discrete barrier option with final maturity date T. At each time T_q, $q = 1, \ldots, Q$, a barrier condition is checked. A payoff H at time T depends upon a set of barrier condition being satisfied. Without loss of generality we can assume that $T_Q < T$ since if $T_Q = T$ the barrier condition can subsumed into the payoff function H.

Let $B = \left\{B_q^l\right\}_{q=1,\ldots,Q,l=0,\ldots,L_q}$ be a set of barrier levels, where for all q we formally set $B_q^0 = 0$, $B_q^{L_q} = \infty$, and we require $B_q^{l_1} < B_q^{l_2}$ for $l_1 < l_2$. For a fixed q, $\left\{B_q^l\right\}_{l=1,\ldots,L_q-1}$ are barrier levels active at time T_q (B_q^0 and $B_q^{L_q}$ are defined for convenience). For $1 \leq P \leq Q$ define $\Delta^P = \{(l_1, \dots, l_P) \mid l_q \in \{1, \dots, L_q\}, \ q = 1, \dots, P\}$, and set Δ^0 to contain just the empty set. A vector $\widehat{\delta} \in \Delta^P$ will represent a set of generalised barrier conditions that have been met by time T_P.

We define a map $\delta = (\delta_1, \dots, \delta_Q) : \Omega \to \Delta^Q$. For $1 \leq q \leq Q$, set $\delta_q(\omega) = l$ if $B_q^{l-1} < S_{T_q}(\omega) \leq B_q^l$. Note that here we have chosen to adjoin B_q^l to the open interval $\left(B_q^{l-1}, B_q^l\right)$. In general we allow ourselves the freedom to adjoin B_q^l to the interval $\left(B_q^l, B_q^{l+1}\right)$ instead. In the former case we say B_q^l is adjoined down and in the latter it is adjoined up.[2]

δ encodes information on what barriers have been hit along the path ω. We suppose that the payoff $H(\delta, S_T)$ at time T to a generalised barrier option depends upon δ and on S_T, so that $c_0 = \mathrm{e}^{-rT}\mathbb{E}_0\left[H(\delta, S_T)\right]$ under the spot measure. $H(\delta, S_T)$ takes the form

$$H(\delta, S_T) = \sum_{\widehat{\delta} \in \Delta^Q} H^{\widehat{\delta}}(S_T)\, \mathrm{I}_{\left\{\delta = \widehat{\delta}\right\}} \tag{1}$$

where $H^{\widehat{\delta}}(S_T)$ depends only on S_T and $\mathrm{I}_{\{\}}$ is the indicator function. There are potentially $\prod_{q=1,\dots,Q} L_q$ different payoff functions depending on which range $\left(B_q^{l-1}, B_q^l\right]$ the asset value S_{T_q} lies in at each time T_q.

For a down and in call option $L_q = 2$ for all q and

$$H^{\widehat{\delta}}(S_T) = \begin{cases} 0, & \widehat{\delta}_q = 2, \text{ for all } q, \\ (S_T - X)_+, & \text{otherwise,} \end{cases} \tag{2}$$

and for an up and out put $L_q = 2$ for all q and

$$H^{\widehat{\delta}}(S_T) = \begin{cases} (X - S_T)_+, & \widehat{\delta}_q = 1, \text{ for all } q, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where in this case we adjoin B_q^l to the interval $\left(B_q^l, B_q^{l+1}\right)$ for all l so that $\delta_q(\omega) = l$ if $B_q^{l-1} \leq S_{T_q}(\omega) < B_q^l$.

For a double knock out put $L_q = 3$ for all q and

$$H^{\widehat{\delta}}(S_T) = \begin{cases} (X - S_T)_+, & \widehat{\delta}_q = 2, \text{ for all } q, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where B_q^1 is adjoined down and B_q^2 is adjoined up.

2.1 The Dirichlet Lattice

Discretise time as $0 = t_0 < \dots < t_N = T$ where, for the moment, we assume the time step $\Delta t = t_{j+1} - t_j$, $j = 0, \dots, N-1$ is a constant. Label nodes on the lattice at time t_j by the pair (j, i), $i = -N_j, \dots, N_j$, where $N_j = jK$ for a

[2] In fact in our context $\left\{\omega \mid S_{T_q}(\omega) = B_q^l\right\}$ is always a set of measure zero, so that whether B_q^l is adjoined up or down is a technical exercise only. In other contexts this may not be the case.

constant integer K. At node (j,i) we suppose that the discrete lattice process $\widehat{S}$ takes the 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 $S_{0,0} = S_0$ is the initial value of the asset, and $z_{j,i} = i\Delta z$ for an increment Δz. Conditional on $\widehat{S}_{t_j} = S_{j,i}$, set $p_k = \Pr\left[\widehat{S}_{t_{j+1}} = S_{j+1,i+k} \mid \widehat{S}_{t_j} = S_{j,i}\right]$ for $k = -K,\ldots,K$. We shall use trinomial branching, setting $K = 1$, $\Delta z = \sqrt{\kappa\Delta t}$ and

$$p_k = \begin{cases} \frac{1}{2\kappa}, & k = \pm 1, \\ \frac{\kappa-1}{\kappa}, & k = 0. \end{cases} \tag{5}$$

Setting $\kappa = 3$ this conditional branching matches the first five moments of $\ln(S_t)$ over the interval Δt.[3]

On the Dirichlet lattice the lattice process is also defined at intermediate times $t_j < t < t_{j+1}$. At these times it is distributed according to the bridge distribution of S. Let $F_{j,j+1}^{i,i+k}(u \mid t) = \Pr\left[\widehat{S}_t \le u \mid \widehat{S}_{t_j} = S_{j,i}, \widehat{S}_{t_{j+1}} = S_{j+1,i+k}\right]$. Write $R_t = \ln\left(\frac{S_t}{S_0}\right)$ and set $\widehat{u} = \ln\left(\frac{u}{S_0}\right)$. Then

$$\begin{aligned} F_{j,j+1}^{i,i+k}(u \mid t) &= \Pr\left[R_t \le \widehat{u} \mid R_{t_j} = \left(r-\frac{1}{2}\sigma^2\right)t_j + \sigma z_{j,i}, R_{t_{j+1}} = \left(r-\frac{1}{2}\sigma^2\right)t_{j+1} + \sigma z_{j+1,i+k}\right] && (6)\\ &= \Pr\left[R_t \le \widehat{u} - \left(r-\frac{1}{2}\sigma^2\right)t_j - \sigma z_{j,i} \mid R_{t_j} = 0, R_{t_{j+1}} = \left(r-\frac{1}{2}\sigma^2\right)\Delta t + \sigma k\Delta z\right] && (7)\\ &= \Pr\left[z_t \le \frac{\widehat{u}}{\sigma} - \left(\frac{r}{\sigma} - \frac{1}{2}\sigma\right)t - z_{j,i} \mid z_{t_j} = 0, z_{t_{j+1}} = k\Delta z\right] && (8) \end{aligned}$$

for a Wiener process z_t. This last conditional distribution is well known. Conditional on z_{t_j} and $z_{t_{j+1}}$, z_t is normally distributed, with mean $\left(z_{t_{j+1}} - z_{t_j}\right)\frac{t-t_j}{\Delta t}$ and variance $\frac{(t-t_j)(t_{j+1}-t)}{\Delta t}$.

2.2 Constructing a Dirichlet Lattice for a Discrete Barrier Option

Given a set of barrier dates $\{T_q\}_{q=1,\ldots,Q}$ we assume that we can find a set of times $0 = t_0 < \ldots < t_N = T$ such that no barrier date coincides with a time t_j and only one barrier date lies in any one interval $[t_j, t_{j+1}]$. For each q there is a j_q such that $t_{j_q} < T_q < t_{j_q+1}$. We call the indexes $J^Q = \{j_q\}_{q=1,\ldots,Q}$ barrier indexes. Set $Q_t = \max\{q \mid T_q < t\}$ to be the number of barrier conditions that have been tested by time t, where $Q_t = 0$ if $T_1 > t$, and set $Q_j = Q_{t_j}$. For $\widehat{\delta} \in \Delta^P$ we say the barrier condition $\widehat{\delta}$ is met up to time T_P if the projection of $\delta(\omega)$ onto Δ^P equals $\widehat{\delta}$.

[3] Later we find it convenient to modify this basic formulation by relaxing the assumptions of a constant time step and of trinomial branching.

2.2.1 Option valuation by forward induction

Let $f_S(S)$ be the density function of S_T, $f_{S,\delta}\left(S,\widehat{\delta}\right)$ the joint density function of S_T and δ, and $f_\delta\left(\widehat{\delta}\mid S\right)$ the density of δ conditional on S_T, all conditioned on S_0. The value c_0 of a discrete barrier option at time t_0 is

$$\begin{aligned} c_0 &= \mathrm{e}^{-rT}\mathbb{E}_0\left[H\left(\delta,S_T\right)\right] &(9)\\ &= \mathrm{e}^{-rT}\int_0^\infty \sum_{\widehat{\delta}\in\Delta^Q} H^{\widehat{\delta}}(S)\, f_{S,\delta}\left(S,\widehat{\delta}\right)\mathrm{d}S &(10)\\ &= \mathrm{e}^{-rT}\int_0^\infty \sum_{\widehat{\delta}\in\Delta^Q} H^{\widehat{\delta}}(S)\, f_\delta\left(\widehat{\delta}\mid S\right) f_S(S)\,\mathrm{d}S. &(11)\end{aligned}$$

We solve (11) on the lattice.

Write $p_{j,i}$ for the probability on the lattice of reaching node (j,i) from the initial node $(0,0)$ and for $\widehat{\delta}\in\Delta^{Q_j}$ write $p^{\widehat{\delta}}_{j,i}$ for the probability on the lattice of reaching node (j,i) from the initial node $(0,0)$ conditional on the barrier condition $\widehat{\delta}$ being met. $p_{j,i}$ is an approximation on the lattice to $f_S(S_{j,i})$ and $p^{\widehat{\delta}}_{N,i}$ is an approximation to the probability $f_{S,\delta}\left(\widehat{\delta}\mid S_{N,i}\right) f_S(S_{N,i})$. Of course $p_{j,i} = \sum_{\widehat{\delta}\in\Delta^{Q_j}} p^{\widehat{\delta}}_{j,i}$. Write $H^{\widehat{\delta}}_{N,i} = H^{\widehat{\delta}}(S_{N,i})$ for the payoff at node (N,i), conditional on $\delta=\widehat{\delta}\in\Delta^Q$. The forward induction lattice approximation to (11) is

$$c_0 = \mathrm{e}^{-rT}\sum_{i=-N_N}^{N_N}\sum_{\widehat{\delta}\in\Delta^Q} H^{\widehat{\delta}}_{N,i}p^{\widehat{\delta}}_{N,i}. \tag{12}$$

As $N\to\infty$ this discrete approximation converges to its continuous time counterpart.

We evolve forward through the lattice computing $p^{\widehat{\delta}}_{j,i}$ for all $\widehat{\delta}$.[4] Write $\mathcal{B}_{j,i}$ for the set of predecessor nodes to node (j,i),

$$\mathcal{B}_{j,i} = \left\{b\in\{-N_{j-1},\dots,N_{j-1}\}\mid (j-1,b)\ \text{branches to}\ (j,i)\right\}. \tag{13}$$

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

$$p_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{j,b}p_{i-b} \tag{14}$$

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

$\left\{p^{\widehat{\delta}}_{j+1,i}\right\}_{i=-N_{j+1},\dots,N_{j+1}}$ can also be found from $\left\{p^{\widehat{\delta}}_{j,i}\right\}_{i=-N_j,\dots,N_j}$. First suppose that j is not a barrier index so that a barrier condition is not tested over this time step. Then for $\widehat{\delta}\in\Delta^{Q_j}$

$$p^{\widehat{\delta}}_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p^{\widehat{\delta}}_{j,b}p_{i-b}. \tag{15}$$

[4] In practice we need only evolve forward for those values of $\widehat{\delta}$ of interest.

If j is a barrier index, $j = j_q$ for some q, write $F^{i,i+k}_{j,j+1}(q,l)$ for $F^{i,i+k}_{j,j+1}\left(B^l_q \mid T_q\right)$, setting $F^{i,i+k}_{j,j+1}(q,0) = 0$ and $F^{i,i+k}_{j,j+1}(q,L_q) = 1$, and set $\Delta F^{i,i+k}_{j,j+1}(q,l) = F^{i,i+k}_{j,j+1}(q,l) - F^{i,i+k}_{j,j+1}(q,l-1)$. Then for $\widehat{\delta} \in \Delta^{Q_j}$

$$p^{\widehat{\delta}\cup\{l\}}_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} \Delta F^{i,i+b}_{j,j+1}(q,l)\, p^{\widehat{\delta}}_{j,b} p_{i-b} \tag{16}$$

where $\widehat{\delta} \cup \{l\} \in \Delta^{Q_j+1}$ denotes the concatenation of $\{l\}$ onto $\widehat{\delta}$. At time t_0 we have $p^{\widehat{\delta}}_{0,0} = 1$ for $\widehat{\delta} = \{\} \in \Delta^0$. From this starting point one can now evolve $p^{\widehat{\delta}}_{j,i}$ forward through the lattice up to time t_N, and then use them in (12).

Ordinary knock-in and knock-out discrete barrier options can be valued as special cases of (12) and (16). For instance an up and out call, where $L_q = 2$ for all q, only gets a payoff when $\widehat{\delta} = \mathbf{1} = (1, \ldots, 1)$, so

$$c_0 = \mathrm{e}^{-rT} \sum_{i=-N_N}^{N_N} H^{\mathbf{1}}_{N,i} p^{\mathbf{1}}_{N,i}, \tag{17}$$

and $p^{\mathbf{1}\cup\{1\}}_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{i-b} \Delta F^{i,i+b}_{j,j+1}(q,1)\, p^{\mathbf{1}}_{j,b}$ at a barrier index j.

A vanilla up and in call receives a payoff when $\widehat{\delta} \neq \mathbf{1}$, but if a payoff is made it is the same for all $\widehat{\delta} \neq \mathbf{1}$, $H^{\widehat{\delta}}_{N,i} \equiv H^U_{N,i}$ say, so

$$\begin{aligned} c_0 &= \mathrm{e}^{-rT} \sum_{i=-N_N}^{N_N} \sum_{\delta\neq\mathbf{1}} H^{\widehat{\delta}}_{N,i} p^{\widehat{\delta}}_{N,i} && (18) \\ &= \mathrm{e}^{-rT} \sum_{i=-N_N}^{N_N} H^U_{N,i} \sum_{\delta\neq\mathbf{1}} p^{\widehat{\delta}}_{N,i} && (19) \\ &= \mathrm{e}^{-rT} \sum_{i=-N_N}^{N_N} H^U_{N,i} \left(p_{N,i} - p^{\mathbf{1}}_{N,i}\right), && (20) \end{aligned}$$

and $p^{\mathbf{1}\cup\{1\}}_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{i-b} \Delta F^{i,i+b}_{j,j+1}(q,1)\, p^{\mathbf{1}}_{j,b}$ at a barrier index as before.

A double knock in option is treated similarly. We have $L_q = 2$ for all q. Let

$$\begin{aligned} \Delta^{P,U} &= \left\{\widehat{\delta} \in \Delta^P \mid \exists q \leq P \text{ st } \widehat{\delta}_q = 3 \text{ and } \forall q' < q\ \widehat{\delta}_q \neq 1\right\}, && (21) \\ \Delta^{P,L} &= \left\{\widehat{\delta} \in \Delta^P \mid \exists q \leq P \text{ st } \widehat{\delta}_q = 1 \text{ and } \forall q' < q\ \widehat{\delta}_q \neq 3\right\}, && (22) \end{aligned}$$

and set $p^U_{j,i} = \sum_{\widehat{\delta}\in\Delta^{Q_j,U}} p^{\widehat{\delta}}_{j,i}$, $p^L_{j,i} = \sum_{\widehat{\delta}\in\Delta^{Q_j,L}} p^{\widehat{\delta}}_{j,i}$, $p^O_{j,i} = p_{j,i} - p^U_{j,i} - p^L_{j,i}$, so that $p^O_{j,i} = p^{\mathbf{2}}_{j,i}$ where $\mathbf{2} = (2, \ldots, 2)$. These are the probabilities that at node (j,i) the option has knocked in at the upper barrier, the lower barrier, and that it has not yet knocked in, respectively. Then at a barrier index (16) reduces to

$$p^U_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{i-b}\left(p^U_{j,b} + \Delta F^{i,i+b}_{j,j+1}(q,3)\, p^O_{j,b}\right), \tag{23}$$

$$p^L_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{i-b}\left(\Delta F^{i,i+b}_{j,j+1}(q,1)\, p^O_{j,b} + p^L_{j,b}\right), \tag{24}$$

$$p^O_{j+1,i} = \sum_{b\in\mathcal{B}_{j+1,i}} p_{i-b}\Delta F^{i,i+b}_{j,j+1}(q,2)\, p^O_{j,b}, \tag{25}$$

and

$$c_0 = \mathrm{e}^{-rT}\sum_{i=-N_N}^{N_N}\left(H^3_{N,i}p^U_{N,i} + H^2_{N,i}p^O_{N,i} + H^1_{N,i}p^L_{N,i}\right) \tag{26}$$

where $H^l_{N,i}$, $l = 1, 2, 3$ are the payoffs to the options if it has been knocked in at the upper boundary, if it has not been knocked in and the payoff if it has been knocked in at the lower boundary, respectively. For a vanilla double knock in, $H^1_{N,i} = H^3_{N,i}$ and $H^2_{N,i} = 0$ for all (N, i).

2.2.2 Option valuation by backwards induction

Backwards induction needs to be used if a rebate is paid or payable when a 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. Let $\delta_t \in \Delta^{Q_t}$ be the barrier conditions met up to time t and write $H_t(\delta_t, S_t)$ for the payoff to the option if it is exercisable at time $t \leq T$, conditional on the value of δ_t. Write δ_j for δ_{t_j}. An exercise strategy $\sigma \leq T$ is a stopping time at which exercise takes place. In the American version of (9),

$$c_0 = \max_\sigma\left\{\mathbb{E}_t\left[\exp\left(-\int_t^\sigma r_s\mathrm{d}s\right) H_\sigma(\delta_\sigma, S_\sigma)\,\mathrm{I}_{\{\sigma\leq T\}}\right]\right\} \tag{27}$$

where the maximum is taken over all exercise strategies σ. Over an interval $[t, t+\Delta t]$, conditional upon the option not having been exercised by time t, we have

$$c^{\delta_t}_t = \max_\sigma\left\{\mathbb{E}_t\left[\exp\left(-\int_t^\sigma r_s\mathrm{d}s\right) H_\sigma(\delta_\sigma, S_\sigma)\,\mathrm{I}_{\{\sigma\leq t+\Delta t\}} + \exp\left(-\int_t^{t+\Delta t} r_s\mathrm{d}s\right) c^{\delta_{t+\Delta t}}_{t+\Delta t}\mathrm{I}_{\{\sigma>t+\Delta t\}}\right]\right\}. \tag{28}$$

Backwards induction solves (27) by iteration back from time T by discretising (28). To discretise we suppose that exercise is not possible between times t and $t + \Delta t$. Let $v^{\delta_t}_t = \mathbb{E}_t\left[\exp\left(-\int_t^{t+\Delta t} r_s\mathrm{d}s\right) c^{\delta_{t+\Delta t}}_{t+\Delta t}\mathrm{I}_{\{\sigma\geq t+\Delta t\}}\right]$ be the continuation value of the option; the option value at time t if it is not exercised before time $t + \Delta t$ but exercised optimally thereafter. Then

$$\widehat{c}^{\delta_t}_t = \max\left\{H_t(\delta_t, S_t), v^{\delta_t}_t\right\} \tag{29}$$

is an approximation to $c_t^{\delta_t}$. $\widehat{c}_t^{\delta_t}$ converges to $c_t^{\delta_t}$ as $\Delta t \to 0$.

Let $H_{j,i}^{\widehat{\delta}} = H_t^{\widehat{\delta}}(S_{j,i})$ be the payoff to the option if exercised at node (j,i) on the lattice, conditional on $\widehat{\delta} \in \Delta^{Q_j}$. Set $c_{N,i}^{\widehat{\delta}} = H_{N,i}^{\widehat{\delta}}$ and simultaneously evolve back $c_{N,i}^{\widehat{\delta}}$ for all $\widehat{\delta}$. At time step t_j only conditions in the set Δ^{Q_j} are admissible.

If j is not a barrier index then the continuation value on the lattice is

$$v_{j,i}^{\widehat{\delta}} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k c_{j+1,i+k}^{\widehat{\delta}}, \qquad \widehat{\delta} \in \Delta^{Q_j}. \tag{30}$$

If $j = j_q$ is a barrier index then we can give prices to options conditional on the barrier conditions met up to time t_{j_q}. Over the step $[t_j, t_{j+1}]$ the state variable satisfies the lth barrier condition with probability $\Delta F_{j,j+1}^{i,i+k}(q,l)$. At time $t_{\bar{j}}$

$$v_{j,i}^{\widehat{\delta}} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k \sum_{l=1,\dots,L_q} \Delta F_{j,j+1}^{i,i+k}(q,l)\, c_{j+1,i+k}^{\widehat{\delta}\cup\{l\}}, \qquad \widehat{\delta} \in \Delta^{Q_j}. \tag{31}$$

At exercise times t_j set

$$c_{j,i}^{\widehat{\delta}} = \max\left\{H_{j,i}^{\widehat{\delta}}, v_{j,i}^{\widehat{\delta}}\right\}, \qquad \widehat{\delta} \in \Delta^{Q_j}, \tag{32}$$

otherwise $c_{j,i}^{\widehat{\delta}} = v_{j,i}^{\widehat{\delta}}$.

For a European down-and-out option, with value $c_{N,i} = H_{N,i}$ at time t_N, (31) reduces to

$$c_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k \left(1 - F_{j,j+1}^{i,i+k}(q,1)\right) c_{j+1,i+k}. \tag{33}$$

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 down-and-in call option all knock-in options are the same. Write $c_{j,i}^{\mathrm{L}}$ for their common value at node (j,i), with value $c_{N,i}^{\mathrm{L}} = H_{N,i}^{\mathrm{L}}$ at time t_N. The option value if not knocked in is $c_{N,i} = H_{N,i} = 0$. Equations (30), (31) and (32) reduce to

$$c_{j,i}^{\mathrm{L}} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k c_{j+1,i+k}^{\mathrm{L}}, \qquad j > j_1, \tag{34}$$

$$c_{j,i} = \begin{cases} \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k c_{j+1,i+k}, & j \notin J^Q, \\ \mathrm{e}^{-r\Delta t} \sum_{k=-K,\dots,K} p_k \left(\left(1 - F_{j,j+1}^{i,i+k}(q,1)\right) c_{j+1,i+k} + F_{j,j+1}^{i,i+k}(q,1)\, c_{j+1,i+k}^{\mathrm{L}}\right), & j \in J^Q, \end{cases} \tag{35}$$

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 $c_{j+1,i+k}^{\mathrm{L}}$.

For a double knock in option $L_q = 3$. At node (j,i) let $c_{j,i}^3$ be the value of the option if it has been knock-in at the upper boundary, $c_{j,i}^1$ its value if it has been knocked in at the lower boundary and $c^2_{j,i}$ if it has not yet been knocked in, with respective payoffs $H^l_{j,i}$, $l = 1, 2, 3$. A vanilla double knock would have $H^l_{j,i} = 0$, for $j < N$, $H^2_{N,i} = 0$ and $H^1_{N,i} = H^3_{N,i}$. Then

$$c^3_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c^3_{j+1,i+k}, \qquad j > j_1, \tag{36}$$

$$c^1_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c^1_{j+1,i+k}, \qquad j > j_1, \tag{37}$$

$$c^2_{j,i} = \begin{cases} \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c^2_{j+1,i+k}, & j \notin J^Q, \\ \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \sum_{l=1,2,3} \Delta F^{i,i+k}_{j,j+1}(q,l)\, c^l_{j+1,i+k}, & j \in J^Q. \end{cases} \tag{38}$$

3 Numerical Results

In this section we value various discrete barrier options on an underlying asset following a geometric Brownian motion. No rebates are earned or paid.

We first benchmark to down and out barrier options, comparing our results with those obtained by other methods. We then use the lattice to price American discrete barrier options, barrier options with non-constant barriers, and complex barrier options. We find that the Dirichlet lattice prices more accurately that either a plain lattice method or conditional Monte Carlo.

All lattices, Dirichlet and Plain, are in any case truncated at 8 standard deviations either side of the expected final value of the underlying.

In practice $\Delta F^{i,i+k}_{j,j+1}(q,l)$ is very close to zero if $S_{j_q,i}$ is some distance away from B^l_q. If in (16) and (31) $\Delta F^{i,i+k}_{j,j+1}(q,l)$ is set to zero when node (j_q, i) is more than ten layers beneath B^l_q there is no difference to machine accuracy in the computed option value, but the computation time is noticeable reduced. This method, with a cut-off ten layers from the barrier, is used to compute the results in this section.

The convergence and accuracy of both forward and backwards induction is improved by using a terminal correction. A terminal correction can be used if there exists a good approximate analytical solution for the value of the option. For a discrete barrier option with $T_Q < T$ the option is European for times $T_Q < t \leq T$ and an explicit solution might exist. In our examples this is the case. One then evolves the lattice only up to the time step $j_Q + 1$ immediately following time T_Q. At each node at time t_{j_Q+1} one assigns an option value equal to the analytical approximation. These values are then evolved back in the lattice, or used in (12) for time t_{j_Q+1}.

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 (2000)). Note that separate terminal corrections need to be made to each component option $c^{\widehat{\delta}}_{j,i}$.

We found significant improvements in convergence thought the use of a terminal correction. Consequently all the results of this section were computed using a terminal correction (except where otherwise stated). .

For our numerical examples we assume that reset times are equally spaced, $T_q = \frac{q}{Q}T$, $q = 1, \ldots, Q$, where the barrier condition at time $T_Q = T$ is subsumed into the payoff at time T. We can then define discrete time steps as follows. Let $N = 1 + RQ$ for some integer $R > 0$. Set $\Delta t = T/(N-1)$. Time steps are

$$t_0 = 0,\ t_N = T, \tag{39}$$

$$t_j = \left(j - \frac{1}{2}\right)\Delta t,\ j = 1, \ldots, N-1. \tag{40}$$

The first step and the last step are of length $\frac{1}{2}\Delta t$. We use modified branching at these steps.

If a terminal correction is being used we do not need to define branching over the last time step. If a terminal correction is not being used at the final step we refine the branching by defining nodes (N, i) for $i = -N, -N + \frac{1}{2}, \ldots, N - \frac{1}{2}, N$, The stock value $S_{N,i}$ at node (N, i) is $S_{N,i} = S_{0,0} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma i \sqrt{\kappa \Delta t}\right)$ as before. Branching from node $(N-1, i)$ is to nodes $\left(N, i + \frac{1}{2}k\right)$, $k = -1, 0, 1$, with probabilities p_k given by (5).

We can define a trinomial branching at the first step, branching from $(0,0)$ to nodes $(1,-1)$, $(1,0)$ and $(1,1)$ with probabilities $\widehat{p}_k$, $k = -1, 0, 1$, $\widehat{p}_{-1} = \widehat{p}_1 = \frac{1}{4\kappa}$, $\widehat{p}_0 = \frac{2\kappa - 1}{2\kappa}$. This choice matches the first three moments of z_t. To match five moments, as before, so that the order of convergence is maintained, one would require heptanomial branching (see Alford and Webber (2000)) at this step.

This can be set up as follows. There are $N' = N - 1$ time steps with $t_0 = 0$, $t_{N'} = T$, $t_j = \left(j + \frac{1}{2}\right)\Delta t$ for $j = 1, \ldots, N' - 1$, where $\Delta t = T/(N-1)$ as before. The first time step is now of length $\frac{3}{2}\Delta t$. At time t_j, $1 \le j < N$, nodes are labelled (j, i), $i = -j - 2, \ldots, j + 2$, and $z_{j,i} = i\Delta z = i\sqrt{\kappa \Delta t}$ as before. For $j \ge 1$ branching is trinomial. For $j = 0$ branching is from $(0,0)$ to nodes $(1, k)$ with probabilities p_k for $k = -3, \ldots, 3$. These probabilities can be chosen to match the first five moments of the Wiener process z over the interval $\left[0, \frac{3}{2}\Delta t\right]$.

Although superficially attractive we found that this refinement did not improve the convergence of the method in this instance. Consequently, all results quoted in this section use trinomial branching throughout.

3.1 Benchmark Results

Tables 1, 2 and 3 present benchmark results for down and out barrier call options with $T = 0.2$ and $T = 2$ years to final maturity. There are either 5 or 50 resets and the barrier level $L = B_q^1$ is 85, 91, or 97. The barrier condition is applied at the maturity time. Parameter values are $S_0 = 100$, $r = 0.1$ and $\sigma = 0.3$ with the strike $X = 100$. A terminal correction is imposed at the first time step following the penultimate barrier and trinomial branching is used at the first time step.

Code was written in VBA 6.0 with no special speed-ups. The platform was a 1.8 Ghz Pentium 4 PC. The top value in square brackets is the time in seconds taken by the forwards induction method, the second value is the time taken by backwards induction.

The benchmark values are taken from Broadie *et al.* (1997) who implemented several valuation methods. BGK is the Broadie *et al.* (1997) analytical correction value, HK is the Heynen and Kat (1994) method, BGK + HK is a heuristic compromise given by Broadie *et al.* (1997) between the analytical correction and the HK method, and BGK lattice is the Broadie *et al.* (1999) trinomial lattice quoted in Broadie *et al.* (1997). Comparisons are also given with a plain lattice method. The plain lattice performs better when the barrier is less likely to be hit, that is, when L is further away from S_0 or when there are fewer barriers per unit time, but it is biased, particularly when $L = 97$.

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.

We see that the Dirichlet lattice is valuing the benchmark instrument to at least 3 decimal places, and sometimes more. In particular the Dirichlet lattice confirms the results of the BGK lattice, including the difficult case of $L = 97$ when the barrier is close to the initial value of the asset. It performs significantly better than the plain lattice.

Tables 4 and 5 benchmark against double knock out options. All options mature in $T = 0.5$ years. The lower boundary is at $L = B_q^1 = 95$ and the upper boundary $U = B_q^2$ takes the values 110, 125 or 150. There are either 25 or 50 evenly spaced resets up to and including the final maturity date. A terminal correction is applied. Parameter values are $S_0 = 100$, $r = 0.1$, $\sigma = 0.2$ with the strike $X = 100$.

Benchmark values are taken from Fusai and Recchioni (2003) who implement the Markov chain Monte Carlo method of Duan *et al.* (2003), (DDGS), and two quadrature methods, one using a trapezium rule (FR-T) and the other Simpson's rule (FR-S), amongst others. Results from a plain lattice method are given for comparison. Similar comments apply here as in the single barrier case.

The Dirichlet lattice confirms the results of Fusai and Recchioni. Although it is not possible to make direct comparisons of computation times[5] we believe our times compare well with theirs. The Dirichlet lattice performs better than the plain lattice.

3.2 Application to Non-Vanilla Discrete Barrier Options

We apply the partial Dirichlet lattice using backwards induction to price American up-and-in discrete barrier options, barrier options with non-constant barriers, and a more complex revivable barrier option. We benchmark the European style options with a plain Monte Carlo method. The Monte Carlo method uses exact discretisation and has $100,000$ sample paths with time steps at the reset times. In the section we take $S_0 = 100$, $r = 0.1$ and $\sigma = 0.2$. All barriers are equally spaced with $T_Q = T$.

[5] Fusai and Recchioni implemented their methods in Fortran on a 600 Mhz Pentium II PC.

3.2.1 American Options

Table 6 shows convergence of the method for American up-and-in puts with strike $X = 100$ and $T = 1$ years to maturity, a barrier level $U = B_q^1$ at either 110 or 130, and either $Q = 4$, 10 or 50 reset dates up to $T = 1$. Since the plain lattice method cannot be used to value up-and-in American options, and it is awkward to use Monte Carlo for American options, the table gives no comparisons. No terminal correction is used in this case. 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, $N = 1 + RQ$. Times shown are for $U = B_q^2 = 130$ with 10 resets.

The lattice appears to have converged to 3 and sometimes 4 significant figures. It has similar accuracy, in terms of relative error, over the range of barrier levels and reset frequencies. As expected, the value of the option increases as the number of reset dates increases, and decreases as U increases.

3.2.2 Non-Constant Barrier Options

We value a pair of down and out call with non-constant barriers. Both options have $T = 1$ and $Q = 50$. The first option has stepped barriers, with $B_q^1 = 95$ for $q = 1, \ldots, 25$ and $B_q^1 = 90$ for $q = 26, \ldots, 50$. The second option has non-linear barriers with values $B_q^1 = 95 \exp\left(0.1 t_q - 0.2 t_q^2\right)$.

Table 7 compares results from the plain Monte Carlo method with the Dirichlet lattice. The stepped discrete barrier option appears to be valued to 5 significant figures, the non-linear discrete barrier option to 3 or 4. The barrier for the non-linear barrier option rises from 95 up to about 96.2 at time 0.24, close to the value of S_0, before decreasing, so the lattice is able to price this option less accurately than the stepped barrier option. The Monte Carlo estimates agree with the lattice but have considerable standard error. This would decrease were speed-up to be used, but Monte Carlo is unlikely to give greater accuracy than the lattice in comparable times.

3.2.3 Complex Barrier Options

Barrier option of any degree of complexity may be valued. Consider a revivable double knock-out barrier option. This option knocks out if upper or lower barriers are hit, but if knocked out it can be knocked back again (perhaps with a different payoff) if a third barrier level is hit. As an example suppose $L_q = 4$ and the option knocks out at levels B_q^1 and B_q^2 but is revived at level B_q^3. At node (j, i) write $c_{j,i}$ for the option value, with payoff $H_{N,i} = (S_{N,i} - X)_+$, if not knocked-out, $c_{j,i}^{\mathrm{O}}$ for the value of the option if it has been knocked out but has not yet been knocked in again, with $c_{N,i}^{\mathrm{O}} = H_{N,i}^{\mathrm{O}} = 0$, and $c_{j,i}^{\mathrm{R}}$ for the value of the revived option with payoff $H^{\mathrm{R}}_{N,i} = \left(S_{N,i} - X^{\mathrm{R}}\right)_+$. Then

$$c^{\mathrm{R}}_{j,i} = \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c^{\mathrm{R}}_{j+1,i+k}, \qquad j > j_1, \tag{41}$$

$$c^{\mathrm{O}}_{j,i} = \begin{cases} \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c^{\mathrm{O}}_{j+1,i+k}, & j \notin J^Q, \\ \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \left(\Delta F^{i,i+k}_{j,j+1}(q,4)\, c^{\mathrm{R}}_{j+1,i+k} + \left(1 - \Delta F^{i,i+k}_{j,j+1}(q,4)\right) c^{\mathrm{O}}_{j+1,i+k}\right), & j \in J^Q, \end{cases} \tag{42}$$

$$c_{j,i} = \begin{cases} \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k c_{j+1,i+k}, & j \notin J^Q, \\ \mathrm{e}^{-r\Delta t} \sum_{k=-K,\ldots,K} p_k \left(\left(1 - \Delta F^{i,i+k}_{j,j+1}(q,2)\right) c^{\mathrm{O}}_{j+1,i+k} + \Delta F^{i,i+k}_{j,j+1}(q,2)\, c_{j+1,i+k}\right), & j \in J^Q. \end{cases} \tag{43}$$

We price a revivable double knock-out discrete barrier call option with $T = 0.5$ years and $Q = 25$. Barriers are $B^1_q = 95$, $B^2_q = 110$ or 125 and B^3_q at 5, 10, or 15 above B^2_q. We take $X = 100$ and $X^{\mathrm{R}} = B^3_q$ so that when the option revives it does so at a strike equal to the revivable barrier level.

Table 8 presents convergence of the Dirichlet lattice for this option using backwards induction. Also shown are the values of the comparable non-revivable discrete double knock options (taken from table 4), and the values of discrete up and in calls with barrier at B^3_q (computed with $R = 400$). The value of the discrete down and out option with barrier at $B^1_q = 95$ is 6.63163 (also computed with $R = 400$). For each R times vary little across various specifications, so a single time in seconds, for $\left(B^2_q, B^3_q\right) = (110, 120)$, is given alongside each R in square brackets. The Monte Carlo time also varies little and is given for the same option.

The lattice appears to have converged to 4 or 5 significant figures. The Monte Carlo estimates are consistent with the lattice but are far less precise.

The revivability feature adds value to the non-revivable double knock out option, particularly when the revival level is close to the upper barrier, but the option is still worth significantly less than the discrete down and out. When B^3_q is close to S_0 the revivable option has values close to the up and in option. When B^3_q is further away its values are closer to those of the double knock out.

4 Conclusions

We have presented a framework for valuing complex discrete barrier options and a lattice valuation method based upon a knowledge of the bridge distribution of the underlying asset value. We have benchmarked the lattice to vanilla single and double discrete barrier options and gone on to value American up and in discrete barrier puts, non-constant discrete barrier options, and revivable discrete double knock out options.

We find that the method compares favourably to plain lattice methods and to Monte Carlo.

Although we have presented results only for the case when the underlying asset follows a geometric Brownian motion, the method is applicable more generally to any process for which the conditional bridge distribution is known.

The method is a relatively simple means of obtaining accurate option values for a wide range of complex European and American discrete barrier options.

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] F. Ait-Sahalia and T. L. Lai. Valuation of Discrete Barrier and Hindsight Options. *Journal of Financial Engineering*, 6(2):169–177, 1997.

[3] F. Ait-Sahalia and T. L. Lai. Random Walk Duality and Valuation of Discrete Lookback Options. *Applied Mathematical Finance*, 5(3):227–240, 1998.

[4] J. Alford and N. Webber. Very High Order Lattice Methods for One Factor Models. 2001. working paper, Warwick Business School.

[5] L. Andersen and R. Brotherton-Ratcliffe. Exact Exotics. *Risk*, 9, No.10:85–89, 96.

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

[7] O. Baldi, L. Caramellion, and M. G. Lovino. Pricing General Barrier Options: A numerical Approach Using Sharp Large Deviations. *Mathematic Finance*, 9, No.4:293–322, 99.

[8] 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.

[9] P. Boyle and Y. S. Tian. An Explicit Finite Difference Approach to the pricing of Barrier Options. *Applied Mathematical Finance*, 5:17–43, 1998.

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

[11] M. Broadie, P. Glasserman, and S. Kuo. Connecting Dicrete and Continuous Path-Dependent Options. *Finance and Stochastics*, 3:55–82, 1999.

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

[13] J. C. Duan, E. Dudley, G. Gauthier, and J-G. Simonato. Pricing discretely monitored barrier options by a markov chain. *Journal of Derivatives*, Summer:9–32, 2003.

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

[15] G. Fusai and M. C. Recchioni. Analysis of quadrature methods for the valuation of discrete barrier options. *worling paper, Universita del Piemonte Orientale*, 2003.

[16] S. Heston and G. Zhou. On the Rate of Convergence of Discrete-Time Contingent Claims. *Mathematical Finance*, 00.

[17] R. C. Heynen and H. M. Kat. Partial Barrier Options. *Journal of Financial Engineering*, 3:253–273, 1994.

[18] P. Horfelt. Extension of the corrected barrier approximation by Broadie, Glasserman, and Kou. *Finance and Stochastics*, 7:231–243, 2003.

[19] S. G. Kou. On the Pricing of Discrete Barrier Options. *working paper, Columbia University*, 2001.

[20] G. Kuan and N. J. Webber. Accurate Valuation of Continuous Barrer Options on a Lattice. *working paper, Cass Business School*, 2003.

[21] N. Kunitomo and M. Ikeda. Pricing options with curved boundaries. *Mathematical Finance*, 2(4):271–298, 1992.

[22] E. Reiner and M. Rubinstein. Breaking Down the Barriers. *Risk*, 4(8):28–35, 1991.

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

[24] M. A. Sullivan. Princing Discretely Monitored Barrier Options. *Journal of Computational Finance*, 3(4):35–52, 2000.

[25] Y. Tian. Pricing Complex Barrier Options under General Diffusion Processes. *Journal of Derivatives*, Winter:11–30, 99.

[26] J. Z. Wei. Valuation of Discrete Barrier Options by Interpolation. *Journal of Derivatives*, Fall:51–73, 1998.

[27] R. Zvan, K.R. Vetzal, and P.A. Forsyth. PDE Methods for Pricing Barrier Options. *Journal of Economic Dynamics and Control*, pages 1563–1590, 2000.

Down and Out, $Q = 5$, $T = 0.2$				
L:		85	91	97
BGK:		6.337	6.194	5.028
HK:		6.337	6.205	5.323
BGK + HK:		6.337	6.195	5.141
BGK lattice:		6.337	6.187	5.167
Plain lattice:		6.33688 [4.5]	6.1911 [4.5]	5.1408 [4.5]
R	100	6.33695 [0.14] [0.14]	6.1878 [0.14] [0.13]	5.1706 [0.14] [0.14]
	400	6.33695 [1.2] [1.1]	6.1872 [1.2] [1.1]	5.1685 [1.2] [1.1]
	700	6.33695 [2.8] [2.6]	6.1873 [2.8] [2.6]	5.1682 [2.8] [2.6]
	1000	6.33694 [4.8] [4.5]	6.1874 [4.8] [4.5]	5.1667 [4.8] [4.5]

Table 1: Benchmark: Down and out option, Q = 5, T = 0.2

Down and Out, $Q = 50$, $T = 0.2$				
L:		85	91	97
BGK:		6.322	5.977	3.836
HK:		6.322	5.977	3.845
BGK + HK:		6.322	5.977	3.837
BGK lattice:		6.322	5.977	3.834
Plain lattice:		6.32221 [6.1]	5.9787 [6.2]	3.8370 [6.1]
R	10	6.32218 [0.23] [0.20]	5.9786 [0.22] [0.20]	3.8303 [0.22] [0.20]
	40	6.32224 [1.6] [1.5]	5.9770 [1.6] [1.5]	3.8335 [1.6] [1.5]
	70	6.32225 [3.8] [3.4]	5.9771 [3.8] [3.4]	3.8329 [3.8] [3.4]
	100	6.32224 [6.6] [6.7]	5.9772 [6.5] [6.0]	3.8344 [6.5] [6.0]

Table 2: Benchmark: Down and out option, Q = 50, T = 0.2

Down and Out, $Q = 50$, $T = 2$				
L:		85	91	97
BGK:		20.821	16.446	9.945
HK:		20.822	16.487	10.576
BGK + HK:		20.821	16.450	10.183
BGK lattice:		20.819	16.436	10.254
Plain lattice:		20.8264 [6.1]	16.4238 [6.2]	10.3790 [6.1]
R	10	20.8264 [0.22] [0.22]	16.4315 [0.20] [0.20]	10.2471 [0.22] [0.20]
	40	20.8207 [1.6] [1.5]	16.4346 [1.7] [1.5]	10.2559 [1.6] [1.5]
	70	20.8190 [3.8] [3.5]	16.4333 [3.8] [3.5]	10.2447 [3.8] [3.5]
	100	20.8197 [6.6] [6.3]	16.4362 [6.6] [6.1]	10.2608 [6.5] [6.1]

Table 3: Benchmark: Down and out option, Q = 50, T = 2

Double knock out, $Q = 25$, $T = 0.5$				
U:		110	125	150
DDGS:		0.1630	3.0058	6.2990
FR-T:		0.1630	3.0060	6.2990
FR-S:		0.1630	3.0061	6.2990
Plain lattice:		0.1635 [11.3]	3.0044 [11.3]	6.2930 [11.4]
R	60	0.1630 [1.1] [1.0]	3.0065 [1.1] [1.0]	6.2991 [1.1] [1.0]
	120	0.1629 [3.0] [2.7]	3.0062 [3.0] [2.7]	6.2990 [3.0] [2.7]
	180	0.1630 [5.4] [5.0]	3.0060 [5.5] [5.0]	6.2990 [5.5] [5.0]
	240	0.1630 [8.4] [7.6]	3.0061 [8.4] [67.7]	6.2990 [8.5] [7.7]
	300	0.1630 [11.7] [10.7]	3.0061 [11.7] [10.7]	6.2990 [11.8] [10.7]

Table 4: Benchmark: Double knock out option, Q = 25, T = 0.5

Double knock out, $Q = 125$, $T = 0.5$				
U:		110	125	150
DDGS:		0.0756	2.4802	5.7988
FR-T:		0.0757	2.4818	5.7991
FR-S:		0.0757	2.4818	5.7993
Plain lattice:		0.07561 [12.0]	2.4786 [12.0]	5.7979 [12.0]
R	12	0.07566 [1.3] [1.2]	2.4823 [1.4] [1.3]	5.7997 [1.5] [1.4]
	24	0.07570 [3.4] [3.1]	2.4819 [3.5] [3.2]	5.7989 [3.8] [3.4]
	36	0.07573 [6.0] [5.4]	2.4820 [6.2] [5.6]	5.7993 [6.4] [5.5]
	48	0.07569 [9.4] [8.3]	2.4819 [9.4] [8.5]	5.7991 [9.6] [8.7]
	60	0.07569 [12.7] [11.5]	2.4817 [13.3] [11.7]	5.7992 [13.2] [12.0]

Table 5: Benchmark: Double knock out option, Q = 50, T = 0.5

Up-and-in American puts							
Barrier level:		110			130		
Resets:		4	10	50	4	10	50
N	2501 [5.3]	0.3459	0.5671	0.8491	0.002884	0.005601	0.010927
	5001 [15.4]	0.3448	0.5674	0.8491	0.002879	0.005610	0.010925
	7501 [29.3]	0.3458	0.5682	0.8490	0.002888	0.005614	0.010920
	10001 [44.7]	0.3456	0.5673	0.8491	0.002882	0.005609	0.010922

Table 6: Application: Up-and-in American Puts

Non-Constant Barriers			
Option:		Stepped	Non-linear
Monte Carlo:		11.35 (0.05) [34.6]	8.49 (0.05) [34.1]
R	50	11.37362 [2.1]	8.48364 [2.1]
	100	11.37313 [6.9]	8.48084 [6.8]
	150	11.37316 [11.4]	8.47527 [11.0]
	200	11.37340 [16.7]	8.47609 [17.5]

Table 7: Application: Non-constant barriers

Revivable double knock-out barrier option							
B_q^1:		95					
B_q^2:		110			125		
B_q^3:		115	120	125	130	135	140
Double knock out:		0.16300			3.00627		
Up and In:		2.35377	1.41862	0.82150	0.45834	0.24708	0.12907
Monte Carlo: [17.7]		2.50 (0.02)	1.56 (0.01)	0.97 (0.01)	3.46 (0.02)	3.26 (0.02)	3.14 (0.02)
R	100 [4.4]	2.51674	1.58156	0.98444	3.46407	3.25281	3.13480
	200 [12.3]	2.51680	1.58163	0.98451	3.46461	3.25336	3.13534
	300 [22.3]	2.51680	1.58162	0.98450	3.46452	3.25327	3.13525
	400 [34.6]	2.51681	1.58164	0.98451	3.46455	3.25330	3.13528

Table 8: Application: Revivable double knock-out barrier option