



# **An Efficient Binomial Lattice Method for Step Double Barrier Options**

*Elisa Appolloni, Marcellino Gaudenzi, Antonino Zanette*

November 2012

n. 10/2012

# An Efficient Binomial Lattice Method for Step Double Barrier Options

Elisa Appolloni <sup>\*</sup>, Marcellino Gaudenzi<sup>†</sup>, Antonino Zanette<sup>‡</sup>

## Abstract

We consider the problem of pricing step double barrier options with binomial lattice methods. We introduce an algorithm that is robust and efficient, that treats the 'near barrier' problem for double barrier options and permits the valuation of step double barrier options with American features.

*Keywords:* double barrier options, step double barrier options, American options, tree methods, binomial methods.

## Introduction

Double Barrier options have become quite popular especially in the foreign exchange markets. A double barrier option has a lower barrier and an upper barrier which control the option. Once either of these barriers is breached, the status of the option is immediately determined: either the option comes into existence if the barrier is of knock-and-in type, or ceases to exist if the barrier is of knock-and-out type. Step double barrier options are more flexible contracts that allow investors to set knock-and-out or knock-and-in levels they want. The feature of these contracts is that the double barrier is not constant as in the standard case, but it evolves as a step function of time. Guillame [11] presents closed-form formulae for different types of two-step double barrier options in the Black-Scholes model, but no analytical expressions are given for more general step barrier options. In the latter case the author proposes a conditional Monte Carlo method scheme enhanced with control variate.

In this article we deal with numerical tree methods because they permit to easily treat general multi-step double barrier options, including early exercise features.

The classical CRR approach may be problematic when applied to barrier options because the convergence is very slow compared with the standard case. A possible solution widely shared in literature is to feed the algorithm with the right value of the barrier. In fact the convergence behaviour improves when the barrier lies exactly (or is very close) on a layer of the tree nodes. Boyle-Lau [2] choose the number of time steps in order to minimize the distance

---

<sup>\*</sup> *Dipartimento MEMOTEF, Sapienza Università di Roma, Italy, elisa.appolloni@uniroma1.it*

<sup>†</sup> *Dipartimento di Scienze Economiche e Statistiche, Università di Udine, Italy, gaudenzi@uniud.it*

<sup>‡</sup> *Dipartimento di Scienze Economiche e Statistiche, Università di Udine, Italy, antonino.zanette@uniud.it*

between the barrier and a layer of nodes. Figlewky-Gao [6] introduce an Adaptive Mesh Model that refine the tree mesh near the barrier. Ritchken [13] aligns a layer of nodes of the trinomial tree with each barrier. Later Cheuck-Vorst [3] present a modification of the trinomial method (based on a change of the geometry of the tree) which allows to set a layer of nodes exactly on the barrier for every choice of the number of time steps. Gaudenzi-Lepellere [9] introduce suitable interpolations of binomial values and Gaudenzi-Zanette [12] construct a tree where all the mesh points are generated by the barrier itself. However, all the previous methods are not able to price efficiently double barrier options.

In order to deal with double barrier options Dai-Lyuu [5] introduce the bino-trinomial tree that is constructed so that both barriers exactly hit two lines of the tree nodes. Numerical results show that this method is not able to treat the 'near barrier' problem, occurring when the initial asset price is very close to one of the barriers.

In order to overcome this problem we introduce a method that generates the binomial tree points using the Dai-Lyuu [5] binomial mesh but forgets the trinomial part using just simple interpolations. Moreover, the extension of this method to multi-step double barrier options (including American features) is straightforward and it allows to obtain accurate estimates of the prices in a very short time.

The paper is organized as follows. In Section 1 we present the model and the bino-trinomial tree method for continuous double barrier options and in Section 2 we describe the new proposed lattice algorithm for double barrier options. In Section 3 we provide the extension of this method to step double barrier options. Finally, in Section 4, we compare the results obtained with our algorithm with Dai-Lyuu bino-trinomial tree (double barrier options), Guillaume closed-form formulae (two step double barrier options) and Monte Carlo method (multi-step double barrier options).

## 1 The model and the bino-trinomial method

In this paper, we consider a market model where the evolution of a risky asset is governed by the Black-Scholes stochastic differential equation

$$\frac{dS_t}{S_t} = rdt + \sigma dB_t, \quad S_0 = s_0, \quad (1)$$

where  $(B_t)_{0 \leq t \leq T}$  is a standard Brownian motion under the risk neutral measure  $Q$ . The non-negative constant  $r$  is the force of interest rate and  $\sigma$  is the volatility of the risky asset.

Let  $M$  be the number of steps of the binomial tree and  $\Delta\tau = \frac{T}{M}$  the corresponding time-step. The standard discrete binomial process is given by

$$S_{(i+1)\Delta\tau} = S_{i\Delta\tau} Y_{i+1}, \quad 0 \leq i \leq M - 1,$$

where the random variables  $Y_1, \dots, Y_M$  are independent and identically distributed with values in  $\{d, u\}$ . Let us denote by  $q = \mathbb{P}(Y_M = u)$ . The Cox-Ross-Rubinstein tree corresponds to the choice  $u = \frac{1}{d} = e^{\sigma\sqrt{\Delta\tau}}$  and

$$q = \frac{e^{r\Delta\tau} - e^{-\sigma\sqrt{\Delta\tau}}}{e^{\sigma\sqrt{\Delta\tau}} - e^{-\sigma\sqrt{\Delta\tau}}}.$$

Now, let us consider a continuous double barrier option with barrier levels  $L$  (lower barrier) and  $H$  (higher barrier). In order to treat the double barrier options pricing problem Dai-Lyuu [5] introduce the following bino-trinomial method. After a logarithmic change of the barriers  $l = \log(\frac{L}{s_0})$  and  $h = \log(\frac{H}{s_0})$  they first construct in the log-space a binomial CRR random walk with space step  $\sigma\sqrt{\Delta T}$ , where the new time step  $\Delta T$  is defined as follows.

Considering the CRR choice of time step  $\Delta\tau = \frac{T}{M}$ , the new time step is defined such that

$$\Delta T = \left(\frac{h-l}{2k\sigma}\right)^2$$

where  $k = \lceil \frac{h-l}{2\sigma\sqrt{\Delta\tau}} \rceil$ . By this way, the layers coincide with down barrier  $L$  and up barrier  $H$  and the new number of steps is  $M' = \lfloor \frac{T}{\Delta T} \rfloor$ . Now, it is possible to build a binomial structure of  $M'$  time steps with binomial coefficient  $u = \frac{1}{d} = e^{\sigma\sqrt{\Delta T}}$  and probability  $q = \frac{e^{r\Delta T} - e^{-\sigma\sqrt{\Delta T}}}{e^{\sigma\sqrt{\Delta T}} - e^{-\sigma\sqrt{\Delta T}}}$ . The remaining amount of time to make the whole tree span  $T$  years, that we denote with  $\Delta T'$ , is defined as

$$\Delta T' = T - \left(\left\lfloor \frac{T}{\Delta T} \right\rfloor - 1\right)\Delta T$$

and corresponds to the length of the first time step of the bino-trinomial tree. Finally, Dai-Lyuu construct a 1-step trinomial tree, using a moment matching procedure, starting from  $s_0$  and reaching three nodes of the previous binomial CRR tree at time  $\Delta T'$ . The merge of the binomial tree of  $M'$  steps and the 1-step trinomial tree provide all the mesh structure. The pricing of European or American continuous double barrier options can be done by backward dynamic programming procedure using this bino-trinomial mesh structure. The numerical results in section 4 will show that this binomial-trinomial structure is not able to treat the 'near barrier' problem. In order to overcome this, we introduce a simpler binomial structure called "the binomial lattice" approach.

## 2 The binomial lattice approach for double Barrier options

In the following, we will use the same binomial parameters  $\Delta T$ ,  $u, d$  and  $q$  of Dai-Lyuu [5], computed as described in the previous section. Moreover, we consider a new number of time steps  $N := M' + 2$  in order to perform a suitable interpolation in time.

First of all, we construct a binomial mesh structure where all the binomial nodes are generated by the barriers. Therefore we build a tree which nodes at maturity are indeed all of type

$$Lu^{2j}, \quad j = 0, \dots, k,$$

so that  $Lu^{2k} = H$  (where, as in the previous section,  $k = \lceil \frac{h-l}{2\sigma\sqrt{\frac{T}{M}}} \rceil$ ).

We now proceed to the description of the pricing algorithm in the case of double barrier options knock-and-out. The underlying asset at a generic node  $(i, j)$ ,  $\forall i = 0, \dots, N$ , is

$$S_{i,j} = \begin{cases} Lu^{2j}, & j = 0, \dots, k & \text{if } N - i \text{ is even} \\ Lu^{2j+1}, & j = 0, \dots, k - 1 & \text{if } N - i \text{ is odd} \end{cases}$$

We shall denote by  $v_i(S_{i,j})$  the option price at time  $i$  depending on the underlying  $S_{i,j}$ .

The option prices at maturity are

$$v_N(S_{N,0}) = v_N(S_{N,k}) = 0 \text{ and } v_N(S_{N,j}) = \psi(S_{N,j}), \forall j = 1, \dots, k-1,$$

where  $\psi(x)$  is the payoff function. For call options  $\psi(x) = \max\{x - K, 0\}$ , while for put options  $\psi(x) = \max\{K - x, 0\}$ , where  $K$  is the strike price.

At time step  $i = N - 1, \dots, 0$  it is possible to compute the option price using

$$v_i(S_{i,j}) = e^{-r\Delta T} [qv_{i+1}(S_{i+1,j+1}) + (1-q)v_{i+1}(S_{i+1,j})], \quad j = 0, \dots, k-1, \text{ if } N-i \text{ is odd,}$$

$$v_i(S_{i,j}) = e^{-r\Delta T} [qv_{i+1}(S_{i+1,j}) + (1-q)v_{i+1}(S_{i+1,j-1})], \quad j = 1, \dots, k-1, \text{ if } N-i \text{ is even.}$$

The values at the barriers  $v_i(L)$ ,  $v_i(H)$ , are set equal to 0 at every step  $i$  with  $N - i$  even, in order to take in account the "out" feature of the barrier option.

At time steps  $i = 0$  and  $i = 2$  we choose four nodes (two less and two greater than  $s_0$ ). In order to have a precise price of the double barrier option with respect to the initial value  $s_0$  we interpolate (by a Lagrange 4 points interpolation) these four points at the value  $s_0$  obtaining two "precise" prices at the two different times  $T - (N - 2)\Delta T$  (corresponding at time step  $i = 2$ ) and  $T - N\Delta T$  (corresponding at time step  $i = 0$ ). A linear interpolation at time 0 of these prices will provide the estimation of the option price at time 0 and initial underlying asset  $s_0$ .

We remark that when there are no nodes either between  $s_0$  and  $L$  or between  $s_0$  and  $H$ , we modify the choice of the interpolation points under and over  $s_0$ , taking into account only the three points which are the closest to the barrier. This approach permits us to treat easily and efficiently the 'near-barrier' problem.

In the American case the procedure is similar with suitable differences for the prices values on the barriers. In particular we set  $v_i(L) = \psi(L)$  and  $v_i(H) = \psi(H)$  for each time step  $i = N, \dots, 0$  with  $N - i$  even (see Remark 5.1 in [9]). In the backward procedure, as usual, we need to compare the early exercise with the continuation value at each node of the tree.

The procedure previously described provides an efficient evaluation of double barrier options both in European and American case. We will show this in the last section, concerning the numerical results, where our method will be compared with the bino-trinomial algorithm.

### 3 The binomial lattice approach for step double Barrier options

In this section we apply the previous technique for pricing step double barrier options. Let us introduce the *regular step double barrier options* as explained in Guillaume [11]. Let  $\{t_0, t_1, \dots, t_{n-1}, t_n\}$  be a partition of the option lifetime  $[0, T]$  with  $0 = t_0 < t_1 < \dots < t_n = T$ . A *standard n-step double barrier option* is an option in which the barriers are constant in every interval  $[t_i, t_{i+1}]$ ,  $i = 0, \dots, n - 1$ . Hence, at each interval  $[t_i, t_{i+1}]$  is associated a constant double barrier with down barrier  $L_i$  and up barrier  $H_i$ . A standard n-step double knock-out option with payoff function  $\psi$ , has this payoff at maturity provided that the underlying asset price stayed in  $(L_i, H_i)$  in every interval  $[t_i, t_{i+1}]$ , otherwise it expires worthless or provides a

contractual rebate. It is possible to include in the step double barrier options the possibility to remove the knock-out barrier provision (partial-time step double barrier options). For example an *early ending n-step double knock-out option* with maturity  $t_n$  has the same payoff of a standard call or put on the condition that the underlying asset price stayed in  $(L_i, H_i)$  in every interval  $[t_i, t_{i+1}]$ ,  $i = 0, \dots, n - 2$  (hence there are no "out" condition on the last time interval). A *windows n-step double knock-out option* has the same payoff of a standard call or put on condition that the underlying asset price stayed in  $(L_i, H_i)$  in every interval  $[t_i, t_{i+1}]$ ,  $i = 1, \dots, n - 2$  (hence there are no "out" condition on the first and last time interval).

A partial-time step double barrier option will always be more valuable than the corresponding standard step double barrier option. Moreover, it is possible to take into account knock-in features in all these contracts. In the European case the knock-in options prices are obtained by taking the difference between the prices of the corresponding vanilla option and the knock-out option.

We can apply the binomial lattice approach to treat standard and partial-time step double barrier options in a straightforward way. Let us consider for example a two-step double knock-out option. We globally take  $M$  time steps and we consider  $M_1 = \lfloor \frac{t_1 - t_0}{T} \rfloor$  time steps in the first interval  $[t_0, t_1]$  and  $M_2 = M - M_1$  time steps in the second interval  $[t_1, t_2]$ . We first consider the time period  $[t_1, t_2]$  and we apply the double barrier procedure used in the previous section. So we compute the binomial parameters  $N_2 = M_2' + 2$ ,  $\Delta T_2$ ,  $u_2$ ,  $d_2$ ,  $q_2$ ,  $k_2$  in order to hit exactly the barriers  $L_2$  and  $H_2$ . This leads to a new binomial mesh  $\{S_{i,j}^2\}$  defined  $\forall i = 0, \dots, N_2$  as follows

$$S_{i,j}^2 = \begin{cases} L_2 u_2^{2j}, & j = 0, \dots, k_2 & \text{if } N_2 - i \text{ is even} \\ L_2 u_2^{2j+1}, & j = 0, \dots, k_2 - 1 & \text{if } N_2 - i \text{ is odd} \end{cases}$$

We can then proceed using the backward procedure for  $i = N_2, \dots, 0$  as described in the previous section. With the linear interpolation in time at  $t_1$  we can obtain at every node  $S_{0,j}^2$  the corresponding option price  $v_0^2(S_{0,j}^2)$ .

Now we proceed in the same way in time interval  $[t_0, t_1]$ . We compute the new binomial parameters  $N_1$ ,  $\Delta T_1$ ,  $u_1$ ,  $d_1$ ,  $q_1$ ,  $k_1$  in order to hit exactly the barriers  $L_1$  and  $H_1$ . This leads to a new binomial mesh structure  $\{S_{i,j}^1\}$ . In order to obtain the option prices on the new nodes with underlying  $S_{N_1,j}^1$ ,  $j = 0, \dots, k_1$ , we interpolate at every  $S_{N_1,j}^1$ ,  $j = 0, \dots, k_1$  by a Lagrange interpolation using 4 suitable points in the set  $\{(S_{0,j}^2, v_0^2(S_{0,j}^2))\}$ , with  $j = 0, \dots, k_2$  if  $N_2$  is even and with  $j = 0, \dots, k_2 - 1$  if  $N_2$  is odd. In order to perform such interpolation we set  $v_0^2(S_{0,j}^2) = 0$ , for  $j$  such that  $S_{0,j}^2 \leq L_2$  or  $S_{0,j}^2 \geq H_2$ . Moreover, the values  $v_{N_1}^1(S_{N_1,j}^1)$  will be set equal to zero if either  $S_{N_1,j}^1 \leq L_2$  or  $S_{N_1,j}^1 \geq H_2$ .

Finally, we proceed backward for  $i = N_1, \dots, 0$  and we compute the price at  $s_0$  by a Lagrange interpolation in space and a linear interpolation in time as described before.

In the early ending two-step double knock-out option we just need to add the treatment of the period  $[t_2, t_3]$  where there are no "out" conditions. We start by considering the number of time steps  $M_3$  and the corresponding  $\Delta \tau_3$ . Then we compute  $k_3$  and  $\Delta T_3$  in order to hit exactly the barriers  $L_2$ ,  $H_2$ , i.e.  $k_3 = \lceil \frac{h_2 - l_2}{2\sigma\sqrt{\Delta\tau_3}} \rceil$  and  $\Delta T_3 = \left(\frac{h_2 - l_2}{2k_3\sigma}\right)^2$ . The parameters  $M_3'$ ,  $N_3$ ,  $u_3$ ,  $d_3$ ,  $q_3$  are computed as usual. Now, starting from the nodes evaluated at time  $t_2$  we can consider a tree structure  $\{S_{i,j}^3\}$  in the time interval  $[t_2, t_3]$  of  $N_3$  time steps. At maturity  $t_3$  we obtain

the underlying assets  $S_{N_3,j} = L_2 u_3^j$ ,  $j = -N_3, \dots, 2k_3 + N_3$ . Then we apply the backward CRR binomial procedure starting with the maturity condition at time  $t_3$ . The prices at the nodes  $S_{N_2,j} = L_2 u_2^j$ ,  $j = 0, \dots, k_2$  at time  $t_2$  are obtained with the interpolation in time and space. Now the procedure is the same as in the standard two-step double barrier options.

The treatment of the windows two-steps double knock-out options is similar. In the n-step double barrier options case we just apply the procedure for two-step double barrier options recursively.

## 4 Numerical results

We provide some numerical comparisons of the algorithms presented in the previous sections in the case of double barrier options, two step double barrier options and multi-step double barrier options.

All the computations presented in the tables have been performed in double precision on a PC with a processor Intel Core i5 at 1.7 Ghz.

### 4.1 Double Barrier Options

In order to test the efficiency of the binomial lattice (BL) approach we will consider the numerical experiments proposed in Day-Lyuu [5]. The volatility is  $\sigma = 0.25$ , the interest rate  $r = 0.1$ , the current stock price is  $s_0 = 95$ , the maturity 1 year and the strike price is  $K = 100$ . We consider two cases. In the first case  $L = 90$  and  $H = 140$ . In the second case we consider a case of 'near-barrier':  $L = 94.9$ . We use as benchmark value the closed formula provided by Kunitomo Ikeda [8]. The results are given in Table 1.

M	L=90 H=140			L=94.9 H=140		
	DL	BL	KI	DL	BL	KI
100	1.423589	1.425053		0.274716	0.006089	
200	1.440976	1.441415		0.227618	0.025050	
400	1.449705	1.450196	1.458435	0.109875	0.021563	0.253056
800	1.453358	1.453460		0.128411	0.025215	
1600	1.456403	1.456354		0.062742	0.024799	
3200	1.457183	1.457182		0.053451	0.025114	

Table 1: *Double barrier options prices.*

### 4.2 Two Step Double Barrier Options

Here we will consider the numerical experiments proposed in Guillame [11]. In the European case we use as benchmark value the closed formula provided in [11]. No benchmark is available in the American case. The volatility is  $\sigma = 0.3$ , the interest rate  $r = 0.03$ , the current stock price is  $s_0 = 100$  and the strike price varies:  $K = 90, 100, 110$ .

In Table 2 we report two-step double knock-out put values with double barrier parameters  $t_1 = 0.25$ ,  $t_2 = T = 0.5$ ,  $L_1 = 70$ ,  $H_1 = 130$ ,  $L_2 = 75$ ,  $H_2 = 125$ .

M	K = 90			K = 100			K = 110		
	BL-EU	GUI	BL-AM	BL-EU	GUI	BL-AM	BL-EU	GUI	BL-AM
100	0.806457		3.549508	3.152257		7.694393	7.089792		13.516271
200	0.819128		3.541653	3.186487		7.694650	7.196594		13.574689
400	0.815156	0.821806	3.545095	3.187184	3.194080	7.703400	7.186754	7.186905	13.576866
800	0.820657		3.555169	3.191445		7.712047	7.182939		13.580487
1600	0.821165		3.555965	3.192845		7.713536	7.184522		13.581967
3200	0.821348		3.556271	3.192542		7.713324	7.184678		13.582068

Table 2: *Two-step double knock-out put values.*

In Table 3 we consider an early-ending two step double knock-out call with  $K = 120$  and the double barrier parameters  $t_1 = 0.125$ ,  $t_2 = 0.25$ ,  $t_3 = T = 0.5$ ,  $L_1 = 75$ ,  $H_1 = 125$ ,  $L_2 = 70$ ,  $H_2 = 130$ . The volatility is varying  $\sigma = 0.15, 0.3$ .

M	$\sigma = 0.15$			$\sigma = 0.3$		
	BL-EU	GUI	BL-AM	BL-EU	GUI	BL-AM
100	0.263380		9.962265	1.575926		9.728607
200	0.265933		9.962439	1.613757		9.741826
400	0.271830	0.2755	9.962422	1.605417	1.6165	9.745550
800	0.273739		9.962533	1.608395		9.750609
1600	0.275081		9.962435	1.612841		9.755035
3200	0.275201		9.962400	1.615489		9.758268

Table 3: *Early-ending two step double knock-out call values.*

### 4.3 Multi Step Double Barrier Options

Finally we propose a 16-steps knock out double barrier option. In Table 4 we report 16-steps double knock-out put values with barrier parameters  $t_i = 0.125 \cdot i$ ,  $t_n = T = 2$ ,  $L_i = 70 - i$ ,  $H_i = 130 + i$ ,  $i = 1, \dots, 16$ . The volatility is  $\sigma = 0.3$ , the interest rate  $r = 0.03$ , the current stock price is  $s_0 = 100$  and the strike price is  $K = 110$ . In the European case we use as benchmark value the Monte Carlo method provided in Baldi-Caramellino-Iovino [1] with 10 millions simulations and 1000 Euler time discretization steps (with the confidence interval in parenthesis).

M	BL-EU	MC	BL-AM
100	6.585345		17.570376
200	6.399257		17.598079
400	6.288664	6.197331	17.623151
800	6.243341	[6.187387-6.207276]	17.627720
1600	6.233008		17.629993
3200	6.208183		17.633762
6400	6.203391		17.634432
12800	6.194374		17.635186
25600	6.191878		17.635486

Table 4: *16-step double knock-out put values.*

The computation times are very fast. For example, for  $M = 12800$  and  $M = 25600$  the computation times are 0.028221 and 0.072933 seconds, respectively.



## References

- [1] Baldi, P., Caramellino, L., Iovino, M.G. (1999) : Pricing general barrier options: a numerical approach using sharp large deviations, *Mathematical Finance* **4**, 293-322.
- [2] Boyle, P.P., Lau, S.H. (1994) : Bumping up against the barrier with the binomial method, *The Journal of Derivatives* **1**, 6-14.
- [3] Cheuk, T.H.F., Vorst T.C.F. (1996): Complex barrier options, *The Journal of Derivatives* **4**, 8-22 (1996).
- [4] Cox J., Ross S.A., Rubinstein M. (1979): Option Pricing: A simplified approach, *Journal of Financial Economics* **7**, 229-264.
- [5] Dai, T., Dai Y. (2010): The Bino-Trinomial Tree: A Simple Model for Efficient and Accurate Option Pricing, *The Journal of Derivatives* Vol.17, **4**, 7-24 (2010).
- [6] Figlewski, S., Gao, B. (1999): The Adaptive Mesh model: a New Approach to Efficient Option Pricing, *Journal of Financial Economics* **53**, 313-351.
- [7] Gao, B., Huang. J., Subrahmanyam, M.G. (2000) : The Valuation of American Barrier Options Using the Decomposition Technique, *Journal of Economic Dynamics and Control* Vol.24 **11-12**, 1783-1827.
- [8] Kunitomo M., Ikeda N. (1992) : Pricing options with curved boundaries, *Mathematical Finance* Vol.2, 275-298.
- [9] Gaudenzi, M., Lepellere, M.A. (2006): Pricing and hedging American barrier options by a modified binomial method. *International Journal of Theoretical and Applied Finance* Vol.9, **4** , 533-553.
- [10] Guillame, T. (2003): Window Double Barrier Options, *Review of Derivatives Research*, **6**, 47-75 (2003).
- [11] Guillame, T. (2010): Step Double Barrier Options, *The Journal of Derivatives* **Fall**, 59-79 (2010).
- [12] Gaudenzi, M., Zanette, A. (2009): Pricing American barrier options with discrete dividends by binomial trees, *Decisions in Economics and Finance* Vol.32, **2** , 129-148.
- [13] Ritchken, P. (1995): On pricing barrier options. *The Journal of Derivatives*, **3**, 19-28.