Ch 8. Barrier Option

- Barrier options are path dependent options, but the dependence is weak because we only have to know whether or not the barrier has been triggered during the option life. Any other information associated with the path is not necessary to price barrier options.

- In this chapter, the analytic solution, Monte Carlo simulation, finite difference method, and binomial tree model are introduced to price barrier options. Next, a fast and efficient method to price barrier options based on the reflection principle is introduced. Moreover, some applications of barrier options are shown. Finally, a method to price Parisian option is presented.

I. Analytic Solutions and Monte Carlo Simulation for Barrier Options

- There are closed-form solutions for pricing European-style barrier options. For instance, the pricing formula of the down-and-in call is as follows.

  $c_{\text{down-and-in}} = S_0 \cdot e^{-qt}(B/S_0)^{2\lambda}N(y) - Ke^{-rt}(B/S_0)^{2\lambda-2}N(y - \sigma\sqrt{T})$,

  where $\lambda = \frac{r-q+\sigma^2/2}{\sigma^2}$,

  $y = \frac{\ln[B^2/(S_0K)]}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$,

  B is the barrier level and assumed to be lower than the initial stock price.

- For the eight types of barriers–the combinations of down or up, in or out, and call or put, their closed-form solutions can be referred to Reiner and Rubinstein (1991), "Breaking Down the Barriers," *Risk* 4, pp. 28–35.

- Comparing with developing closed-form solutions, applying the Monte Carlo simulation to pricing barrier options is relatively simple. First, simulate N stock price paths according to $\ln S_{t+\Delta t} \sim ND(\ln S_t + (r - q - \sigma^2/2)\Delta t, \sigma^2 \Delta t)$. Second, decide whether each path is knocked in or out and thus determine the final payoff of each path. Finally, the option value today is the arithmetic average among the present values of the final payoffs of all N stock price paths.

- In addition to pricing standard barrier options, the flexibility of the Monte Carlo simulation is able to deal some exotic features in barrier options, e.g., the discrete-sampling barrier option or the soft barrier option mentioned in Ch 3.

- However, the Monte Carlo simulation works only for European-style barrier options. So, in this chapter, the finite difference method and the binomial tree model are also introduced to price both American- and European-style barrier options.

II. Finite Difference Method to Price Barrier Options

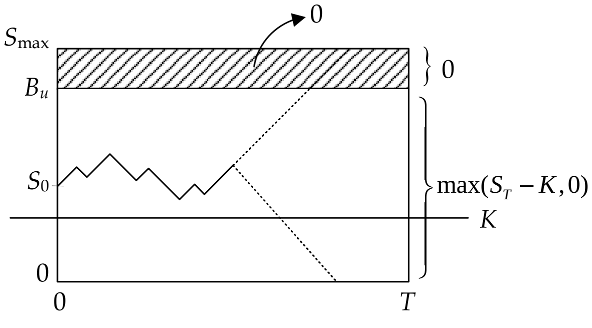
- For different kinds of options, the corresponding partial differential equations are the same to be $\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - q)\frac{\partial V}{\partial S} = rV$. Different options can be priced based on different assumptions of boundary conditions. The boundary conditions for various standard barrier call options are discussed as follows. The extention to barrier put options is straightforward.

  (i) "up-and-out" call (B_u denotes the upper barrier, and $B_u > K$ is assumed)

  $V(S, t) = 0$, for $t < T$ and $S \geq B_u$

  $$V(S,T) = \begin{cases} \max(S - K, 0) & \text{if } S < B_u \\ 0 & \text{o/w} \end{cases}$$

Figure 8-1

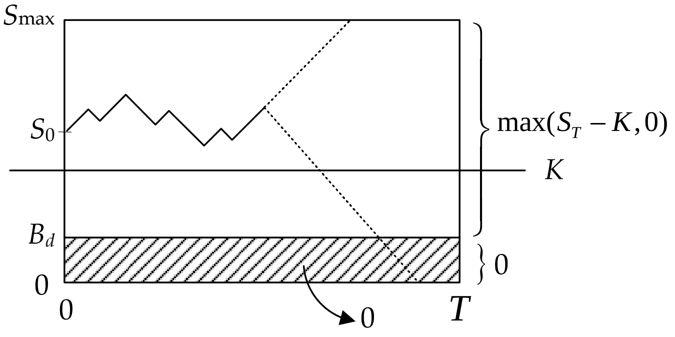


(ii) "down-and-out" call (B_d denotes the lower barrier, and $B_d < K$ is assumed)

$V(S,t) = 0$, for $t < T$ and $S \leq B_d$

$$V(S,T) = \begin{cases} \max(S-K,0) & \text{if } S > B_d \\ 0 & \text{o/w} \end{cases}$$

Figure 8-2



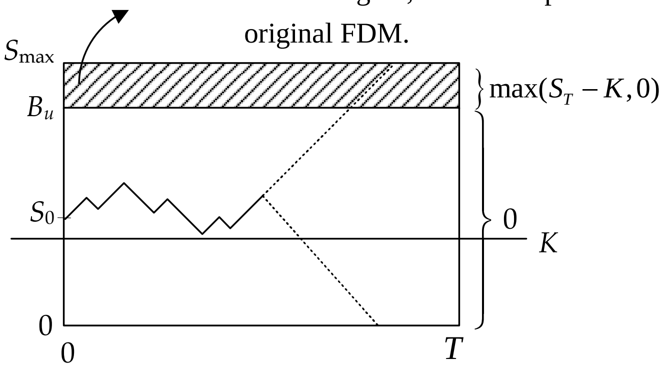
(iii) "up-and-in" call (B_u denotes the upper barrier, and $B_u > K$ is assumed)

$V(S,t) = c(S,t)$, for $t < T$ and $S \geq B_u$

$$V(S,T) = \begin{cases} \max(S-K,0) & \text{if } S \geq B_u \\ 0 & \text{o/w} \end{cases}$$

Figure 8-3

$c(S,t)$ ⇒ It is necessay to calculate the vanilla call option with the FDM to derive $c(S,t)$ for this region, so the computational time of this method is doubled than the original FDM.

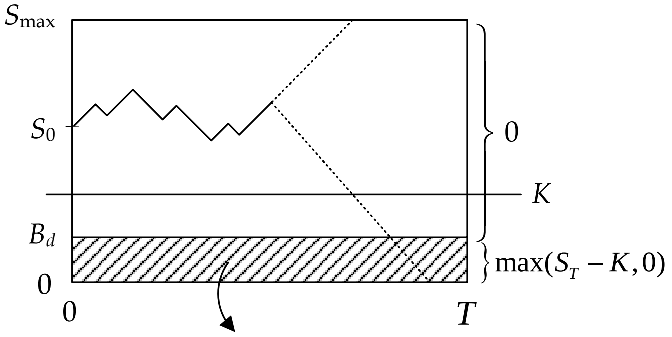


(iv) “down-and-in” call (B_d denotes the upper barrier, and $B_d < K$ is assumed)

$V(S,t) = c(S,t)$, for $t < T$ and $S \leq B_d$

$$V(S,T) = \begin{cases} \max(S-K,0) & \text{if } S \leq B_d \\ 0 & \text{o/w} \end{cases}$$

Figure 8-4



$c(S,t)$ ⇒ It is necessay to calculate the vanilla call option with the FDM to derive $c(S,t)$ for this region, so this method needs double time than the original FDM.

⊙ For (iii) and (iv), the option holder obtains a plain vanilla call option whenever the stock price touches the barrier.

- In-out parity:

European vanilla option = European knock-out option + European knock-in option

⊙ Note the above equation holds only for European options and for the barriers in knock-out and knock-in options being the same.

⊙ For the American versions, since the early exercise time point of an in barrier option may not be the same as that of the corresponding out barrier option, the parity relation does not hold.

⊙ Suppose an investor holds both a knock-out option and a knock-in option. If the stock price never touches the barrier, the knock-out option will provide the same payoff as that of a vanilla option at maturity. When the stock price touches the barrier, the knock-out option becomes worthless and the knock-in option are enabled and providing the same payoff as that of a vanilla option at maturity.

⊙ Since the combined payoff of a knock-out option and a knock-in option are the same as that of a vanilla option at maturity, their values today should be equal. This result is termed as “in-out parity.”

⊙ Alternative method to price knock-in options:

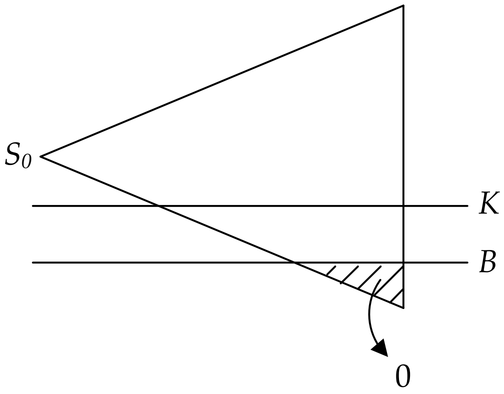
Since it needs more computational effort to price a knock-in option, it is possible to price a knock-out option with the same barrier first, and then apply the in-out parity to derive the value of the knock-in option.

III. Binomial Tree Model to Price Barier Options

- Binomial Tree for barrier options (taking the down-and-out call with $B < K$ as an example)

  ⊙ The straightforward method is to replace the option value with 0 in the region lower the barrier B during the backward induction process.

Figure 8-5

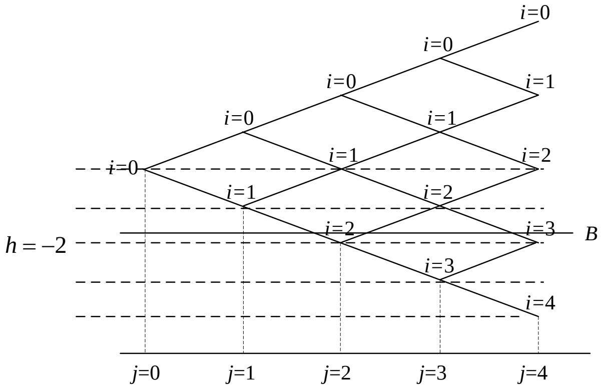


⊙ In fact, it is able to derive correct results if only the option values of the node on the stock price layer just smaller or equal to the barrier are replaced with zero. Following this rule, a more concise and efficient algorithm can be developed on the next page.

- The pricing algorithm in *Financial Engineering and Computation*, by Lyuu.

```
input:  $S, u, d, K, B(B < K, B < S), n, \hat{r} = rT/n = r\Delta t$;
real    $R, p, c[n+1]$;
integer $i, j, h$;
$R := e^{\hat{r}}; p := (R-d)/(u-d)$; //CRR binomial tree model
$h := \lfloor \ln(B/S)/\ln u \rfloor$; //$h$ is the index of the stock price layer just smaller or equal to $B$
for $(i = 0$ to $n)$ $\{c[i] := \max(0, Su^{n-i}d^i - K);\}$ //payoffs for terminal nodes
if $[n-h$ is even and $0 \leq (n-h)/2 \leq n]$
    $c[(n-h)/2] := 0$; //reaching node $(n-h)/2$ means a hit for the barrier
for $(j = n-1$ down to $0)\{$
    for $(i = 0$ to $j)$
        $c[i] := (p \times c[i] + (1-p) \times c[i+1])/R$; //backward induction based on one column vector
     if $[j-h$ is even and $0 \leq (j-h)/2 \leq j]$
        $c[(j-h)/2] := 0$; //reaching node $(j-h)/2$ means a hit for the barrier
}
return  $c[0]$;
```

Figure 8-6



The values of $n = 4$ and $h = -2$ are taken as an example to trace the above program.

⊙ $n = 4$: Since $n - h = 6$ is even and $\frac{n-h}{2} = 3 < n = 4$, the option value for node $\frac{n-h}{2} = 3$ at $j = n$ is set to be zero.

⊙ $j = 3$: Since $j - h = 5$ is odd, it is not necessary to do anything. (Since we consider the down-and-out call, the option value for node 3 at $j = 3$ should be zero. However, the option value for that node does not affect the option value today, because the option value of its parent node, node 2 at $j = 2$, will be set to be zero during the following backward induction procedure.)

⊙ $j = 2$: Since $j - h = 4$ is even and $\frac{j-h}{2} = 2 \leq j = 2$, the option value for node $\frac{j-h}{2} = 2$ at $j = 2$ is set to be zero.

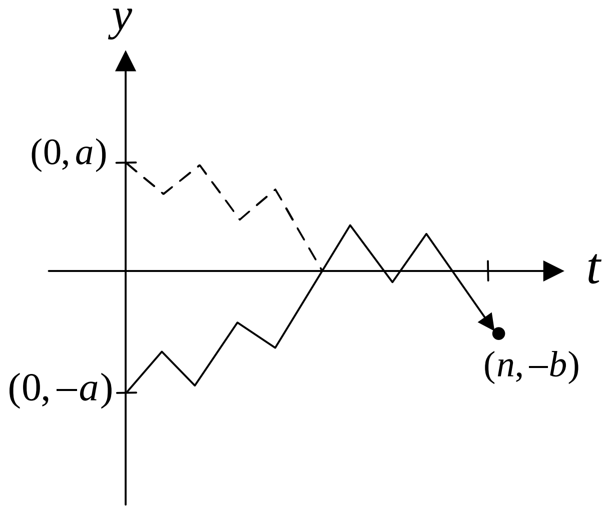
⊙ $j = 1$: Since $j - h = 3$ is odd, it is not necessary to do anything.

⊙ $j = 0$: Since $j - h = 2$ is even and $\frac{j-h}{2} = 1 > j = 0$, it is not necessary to do anything.

IV. Reflection Principle and Pricing Barrier Options

- Mathematical fundamental of the reflection principle:

Figure 8-7



⊙ For all paths from node$(0,-a)$ to node$(n,-b)$, how many of them will touch or cross the x-axis?

In the above figure, the reflection principle states that any path from node$(0,a)$ to node$(n,-b)$ corresponds to one of this kind of paths. So the problem is changed to count the number of paths from node$(0,a)$ to node$(n,-b)$.

⊙ The method to count the nubmer of paths from node$(0,a)$ to node$(n,-b)$:

There are all n time periods, and for each time period, the stock price can move upward or downward: up + down $= n$

In order to start from the level of $y = a$ to the level of $y = -b$, the number of net downward steps should be $a+b$, i.e., down $-$ up $= a+b$

$$\Rightarrow \left\{ \begin{array}{l} \text{up} + \text{down} = n \\ \text{down} - \text{up} = a+b \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \text{down} = \frac{n+a+b}{2} \\ \text{up} = \frac{n-a-b}{2} \end{array} \right.$$

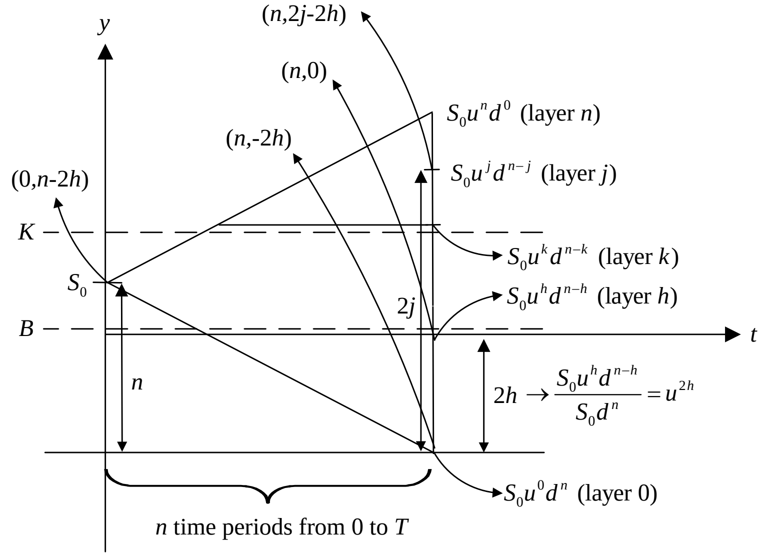
Thus the number of paths from node$(0,a)$ to node$(n,-b)$ is $\binom{n}{\frac{n+a+b}{2}}=\binom{n}{\frac{n-a-b}{2}}$.

As a consequence, the number of the paths from node$(0,-a)$ to node$(n,-b)$ and touching or crossing the x-axis equals $\binom{n}{\frac{n+a+b}{2}}=\binom{n}{\frac{n-a-b}{2}}$.

⊙ Note that $\binom{n}{k}$, the combination of k from n, returns zero for $k<0$, $k>n$, or k is not an integer.

- Apply the reflection principle to pricing down-and-in calls (pp. 234–242 in *Financial Engineering and Computation*, by Lyuu)

  Figure 8-8



$(n,2j\text{-}2h)$
y
$(n,0)$
$S_0u^nd^0$ (layer n)
$(n,\text{-}2h)$
$(0,n\text{-}2h)$
$S_0u^jd^{n-j}$ (layer j)
K
$S_0u^kd^{n-k}$ (layer k)
S_0
$S_0u^hd^{n-h}$ (layer h)
$2j$
B
t
n
$2h \rightarrow \frac{S_0u^hd^{n-h}}{S_0d^n} = u^{2h}$
$S_0u^0d^n$ (layer 0)
n time periods from 0 to T

⊙ First step: index the nodes on the binomial tree and decide the index of the stock price layer for terminal nodes.

$k = \lceil \frac{\ln(K/Sd^n)}{\ln(u/d)} \rceil$, where $\lceil \quad \rceil$ is the upper Gauss operator

· The layer k is the layer just above the strike price.

· For calls, from the layer k to the layer n, these stock price levels are in the money.

$h = \lfloor \frac{\ln(B/Sd^n)}{\ln(u/d)} \rfloor$, where $\lfloor \quad \rfloor$ is the lower Gauss operator

· The layer h is the layer just below the barrier.

· In contrast, if an upper barrier is considered, the layer h just above the barrier is chosen to insure that paths touching or crossing the layer h must touch or cross the barrier.

⊙ Second step: shift the tree on the yt-plane such that the layer h coincides with the t-axis. As a consequence, the root node is indexed as node$(0, n-2h)$ and the node with the stock price $S_0u^jd^{n-j}$ is indexed as node$(n, 2j-2h)$.

⊙ Third step: only the paths from node$(0, n-2h)$ to node$(n, 2j-2h)$ and touching or crossing the x-axis are knocked in and thus with the payoff of $\max(S_0u^jd^{n-j} - K, 0)$.

⇒ According to the reflection principle, the number of knocked-in paths from node$(0, n-2h)$ to node$(n, 2j-2h)$ is $\binom{n}{\frac{n+(n-2h)+(2j-2h)}{2}} = \binom{n}{n-2h+j}$.

⊙ Finally, since we know that each path is with the probability to be $p^j(1-p)^{n-j}$, option value equals

$$e^{-rT} \cdot \sum_{j=k}^{n} \left(\begin{array}{c} n \\ n-2h+j \end{array} \right) \cdot p^j(1-p)^{n-j} \cdot (Su^jd^{n-j} - K).$$

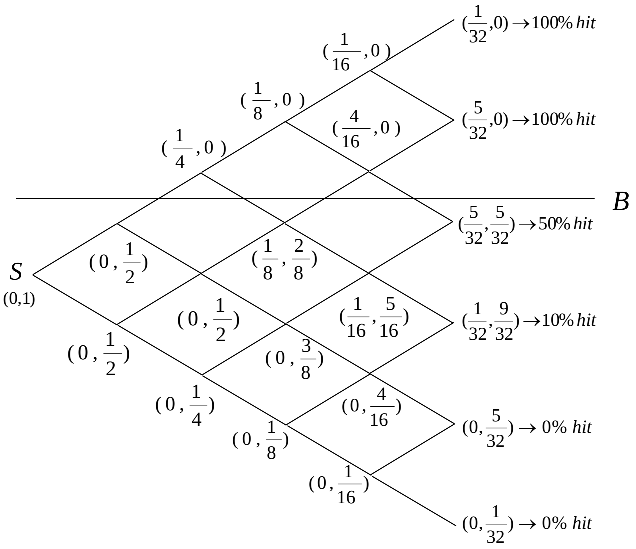
(This method is in essence a combinatorial method, so it is not necessary to build the binomial tree, and down-and-in calls can be priced based on the above equation directly. Note this method or other combinatorial methods works only for European options.)

- Forward-tracking method: to record (or inherit) some information during the forward tree-building phase.

  For barrier options: record (or inherit) the probability pair (hit, not hit).

  The algorithm is to inherit the probability pair from the parent nodes, and reclassify all probability to be the hit probability whenever crossing the barrier from below. In addition, suppose $p = 1 - p = \frac{1}{2}$ for simplicity.

Figure 8-9



S
(0,1)
B
$(\frac{1}{4},0)$
$(0,\frac{1}{2})$
$(0,\frac{1}{2})$
$(\frac{1}{8},0)$
$(\frac{1}{8},\frac{2}{8})$
$(0,\frac{1}{2})$
$(0,\frac{1}{4})$
$(\frac{1}{16},0)$
$(\frac{4}{16},0)$
$(\frac{1}{16},\frac{5}{16})$
$(0,\frac{3}{8})$
$(0,\frac{1}{8})$
$(0,\frac{4}{16})$
$(0,\frac{1}{16})$
$(\frac{1}{32},0)$ → 100% hit
$(\frac{5}{32},0)$ → 100% hit
$(\frac{5}{32},\frac{5}{32})$ → 50% hit
$(\frac{1}{32},\frac{9}{32})$ → 10% hit
$(0,\frac{5}{32})$ → 0% hit
$(0,\frac{1}{32})$ → 0% hit

⇒ The value for a up-and-in call =

$$e^{-rT} \sum_{j=0}^{n} \left(\begin{array}{c} n \\ j \end{array} \right) \cdot p^j \cdot (1-p)^{n-j} \cdot \frac{\text{hit}}{\text{hit+not hit}} \cdot \max(Su^jd^{n-j} - K, 0),$$

where the product of $\left(\begin{array}{c} n \\ j \end{array} \right)$ and $\frac{\text{hit}}{\text{hit+not hit}}$ equals the number of knock-in paths conditional on starting from the root and reaching node(i, j).

(In this method, it is necessary to build a tree and to record (or inherit) the information of the probability pair (hit, not hit) during the forward tree-building phase. However, the advantage of this method is that it is possible to extend this method to pricing American barrier options after proper modification.)

V. Some Applications of Barrier Options

- Since there is a exercise boundary for an American option. American options can be priced as a knocked-and-exercised barrier options.

  ⊙ The exact value of an American put satisfies the following equation.

$$\begin{aligned} P_{Am} &= P_{Eu}(S_t) + \int_t^T rKe^{-r(\tau-t)}N(-d_2(S_t, B_\tau, \tau - t))d\tau \\ & \quad - \int_t^T qS_t e^{-r(\tau-t)}N(-d_1(S_t, B_\tau, \tau - t))d\tau \end{aligned}$$

> The second and third terms represent the interest income and the loss of dividends from the early exercise of the American put. The explanation for these two terms is in Figure 8-10.

$$\begin{aligned} &= P_{Eu}(S_t) + K(1 - e^{-r(T-t)}) - S_t(1 - e^{-q(T-t)}) \\ & \quad -K\int_t^T re^{-r(\tau-t)}N(d_2(S_t, B_\tau, \tau - t))d\tau \\ & \quad +S_t\int_t^T qe^{-q(\tau-t)}N(d_1(S_t, B_\tau, \tau - t))d\tau, \end{aligned}$$

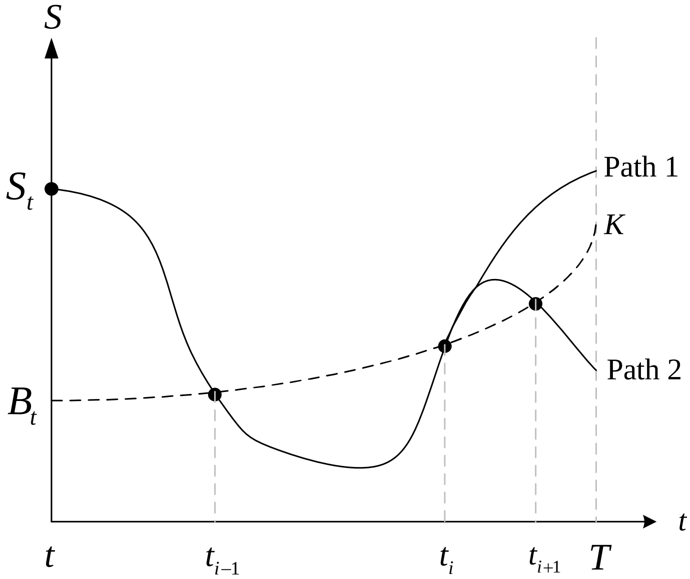
where $d_1(x, y, s) = \frac{\ln(x/y)+(r-q+\sigma^2/2)s}{\sigma\sqrt{s}}$

$d_2(x, y, s) = \frac{\ln(x/y)+(r-q-\sigma^2/2)s}{\sigma\sqrt{s}} = d_1(x, y, s) - \sigma\sqrt{s}$

B_t is the early exercise barrier satisfying the following equation

$$\begin{aligned} K - B_t = P_{Eu}(B_t) &+ K(1 - e^{-r(T-t)}) - B_t(1 - e^{-q(T-t)}) \\ &-K\textstyle\int_t^T re^{-r(\tau-t)}N(d_2(B_t, B_\tau, \tau - t))d\tau \\ &+B_t\textstyle\int_t^T qe^{-q(\tau-t)}N(d_1(B_t, B_\tau, \tau - t))d\tau \end{aligned}$$

Figure 8-10 The exercise boundary of a put option when $r \geq q$



$*$ Note that the exercise boundary of a put option at maturity is $B_T = K$ if $r \geq q$. For the case of $r < q$, $B_T = (r/q)K$. Please refer to Kim (1990), "The Analytic Valuation of American Options," *Review of Financial Studies* 3, pp. 547–572.

$$1.\ \text{At } t_{i-1}, \begin{cases} \text{1. by exercising the American put, short } S_{t_{i-1}} \text{ for cash } K. \\ \text{2. deposit } K \text{ to earn } r \text{ with continuously compounding.} \end{cases}$$

2. During t_{i-1} and t_i, pay dividend yield continuously to the one who owns the stock share.

$$3.\ \text{At } t_i \begin{cases} \text{1. the interest and principal of the deposited cash } K \text{ is } Ke^{r(t_i - t_{i-1})}. \\ \left.\begin{array}{l} \text{2. buy } S \text{ back with } B_{t_i}. \\ \text{3. buy the American put back with } K - B_{t_i} \text{ dollars.} \end{array}\right\} \text{total cost is } K \text{ dollars} \end{cases}$$

$*$ At maturity, it is apparent that this strategy can generate the same payoff as that of a European put, i.e., $\max(K - S_T, 0)$.

∘ For path 1, the stock price will be no longer lower than the exercise boundary, and the final payoff of the American put is $\max(K - S_T, 0) = 0$.

∘ For path 2, the stock price falls below the exercise boundary at t_{i+1}, so we will adopt steps 1 and 2 in the above strategy again. At maturity, buy S back with S_T, and buy the American put back with $K - \mathrm{S}_T$ dollars. The total cost is still K dollars. In addition, the investor can exercise the buy-back American put to earn a positive payoff $\max(K - S_T, 0) = K - S_T$ because the put is in the money at maturity.

$*$ Therefore, the cumulative interest gains and the dividend yield costs for each time point when the stock price falls below B_t construct the early exercise premium.

⊙ Pricing formula for down-and-exercised put with the exponential barrier $Le^{-a(T-t)}$

$$
\begin{aligned}
P_t^E(S_t, L, a) = {} & K[\hat{\lambda}_t^{\hat{b}-\hat{\beta}} N(\hat{d}_0) + \hat{\lambda}_t^{\hat{b}+\hat{\beta}} N(\hat{d}_0 + 2\hat{\beta}\sigma\sqrt{T-t})] \\
& -Le^{-a(T-t)}[\hat{\lambda}_t^{\tilde{b}-\tilde{\beta}} N(\tilde{d}_0) + \hat{\lambda}_t^{\tilde{b}+\tilde{\beta}} N(\tilde{d}_0 + 2\tilde{\beta}\sigma\sqrt{T-t})] \\
& +Ke^{-r(T-t)}[N(\hat{d}_1^-(Le^{-a(T-t)})) - N(\hat{d}^-(K)) + \hat{\lambda}_t^{\hat{\gamma}-1}[N(\hat{d}^+(K)) - N(\hat{d}_1^+(Le^{-a(T-t)}))]] \\
& -S_t e^{-q(T-t)}\{N(\hat{d}_1^-(Le^{-a(T-t)}) + \sigma\sqrt{T-t}) - N(\hat{d}^-(K) + \sigma\sqrt{T-t}) + \\
& \hat{\lambda}_t^{\hat{\gamma}+1}[N(\hat{d}^+(K) + \sigma\sqrt{T-t}) - N(\hat{d}_1^+(Le^{-a(T-t)}) + \sigma\sqrt{T-t})]\}
\end{aligned}
$$

where

$\hat{\lambda}_t = Le^{-a(T-t)}/S_t, \hat{b} = (r - q - a - \frac{1}{2}\sigma^2)/\sigma^2, \hat{\beta} = \sqrt{\hat{b}^2 + 2r/\sigma^2},$

$\hat{d}_0 = \frac{1}{\sigma\sqrt{T-t}}[\ln(\hat{\lambda}_t) - \hat{\beta}\sigma^2(T-t)], \tilde{b} = (r - a - q - \frac{1}{2}\sigma^2)/\sigma^2,$

$\tilde{\beta} = \sqrt{\tilde{b}^2 + 2(r-a)/\sigma^2}, \tilde{d}_0 = \frac{1}{\sigma\sqrt{T-t}}[\ln(\hat{\lambda}_t) - \tilde{\beta}\sigma^2(T-t)], \hat{\gamma} = 2(r-q-a)/\sigma^2,$

$\hat{d}^{\pm}(x) = \frac{1}{\sigma\sqrt{T-t}}[\pm\ln(\hat{\lambda}_t) + \ln(L) - \ln(x) + (r - q - a - \frac{1}{2}\sigma^2)(T-t)],$

$\hat{d}_1^{\pm}(x) = \frac{1}{\sigma\sqrt{T-t}}[\pm\ln(\hat{\lambda}_t) + \ln(L) - \ln(x) + (r - q - 2a - \frac{1}{2}\sigma^2)(T-t)].$

∗ Since the $Le^{-a(T-t)}$ is not the true exercise boundary, if the investor follows this non-optimal exercise strategy, the derived option value will smaller than the true American value, which is priced based on the truly optimal exercise boundary. As a consequence, the option value of a down-and-exercised put is a lower bound for the an American option.

∗ If you can find the L^* and a^* to maximize the option value of the down-and-exercised put, you can derive a very tight lower bound of the American option value, usually within several cents, i.e.,

$$P_{Am} \geq (\approx) \max_{L,a} P_t^E(S_t, L, a)$$

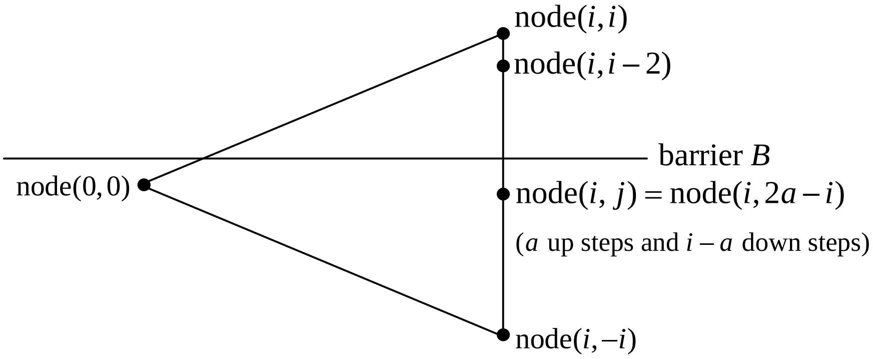
Appendix A. Pricing Parisian Options

- The content in this appendix belongs to the advanced content.

- Apply the combinatorial method to pricing Parisian option. Parisian up-and-out calls are considered, and when the stock price remains above a barrier continuously for a period larger than or equal to a pre-specified time interval (l periods of time in the following example), the option is knocked out.

  Costabile, (2002), "A Combinatorial Approach for Pricing Parisian Options," *Decisions in Economics and Finance* 25, pp. 111–125.

  Figure 8-11 The indexing rule for nodes on the binomial tree.

  
  node(i, i)
node$(i, i-2)$
barrier B
node$(0,0)$
node(i, j) = node$(i, 2a-i)$
(a up steps and $i-a$ down steps)
node$(i, -i)$

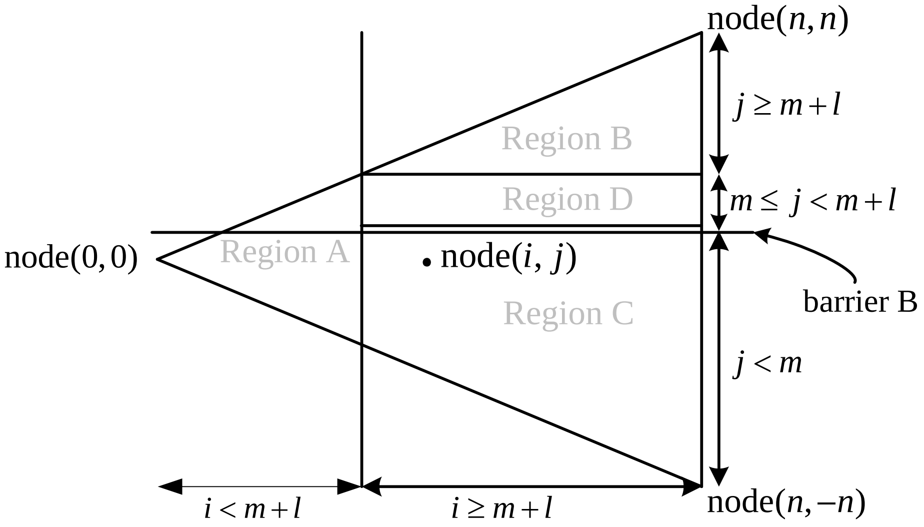
  $N(i,j)$: the number of paths from the origin to node(i,j).

  $g(i,j)$: the number of paths from root to node(i,j) that are characterized by a maximum sojourn time above the barrier B strictly smaller than the time interval of length l, i.e., the still alive paths reaching node(i,j).

  (Since we are pricing up-and-out Parisian options, for any node(i,j), only these $g(i,j)$ paths are not knocked out and only these $g(i,j)$ paths could contribute positive payoff for the option at maturity.)

  m: minimum number of successive up steps that the initial stock price must take to touch or cross the barrier.

Figure 8-12



Region A $(i < m + l)$: $g(i, j) = N(i, j)$ (For all paths reaching this region, the maximum sojourn time above the barrier B is smaller than l.)

Region B $(i \geq m + l, j \geq m + l)$: $g(i, j) = 0$ (A path starting from the root and reaching any node in this region must spends at least l successive periods above the barrier.)

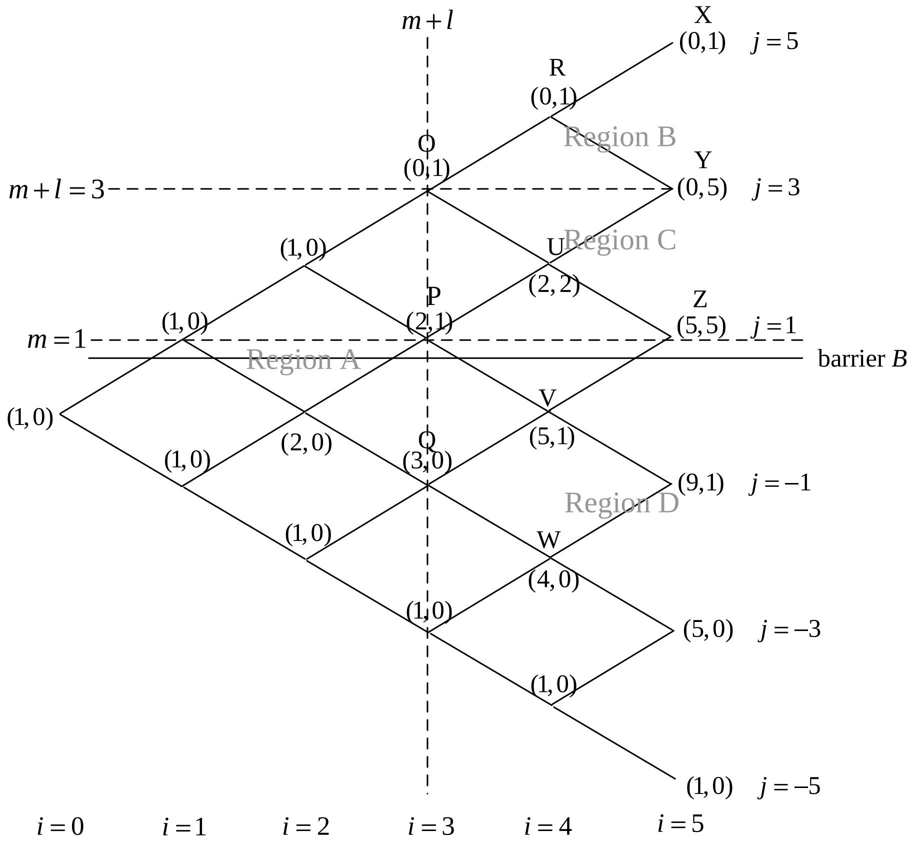
Region C $(i \geq m + l, j < m)$: $g(i, j) = g(i - 1, j + 1) + g(i - 1, j - 1)$ (For any node(i, j) in this region, inherit $g(\cdot, \cdot)$ from its parents nodes. Since node(i, j) is below the barrier, the paths reaching its parent nodes and not knocked out will not be knocked out even after one additional step to node(i, j).)

Region D $(i \geq m + l, m \leq j < m + l)$:

$$g(i, j) = \sum_{0 \leq k < \frac{l-(j-m)}{2}} g(i-(j-m)-2k-1, m-1) \cdot \left[\begin{pmatrix} j - m + 2k \\ j - m + k \end{pmatrix} - \begin{pmatrix} j - m + 2k \\ j - m + k + 1 \end{pmatrix} \right]$$

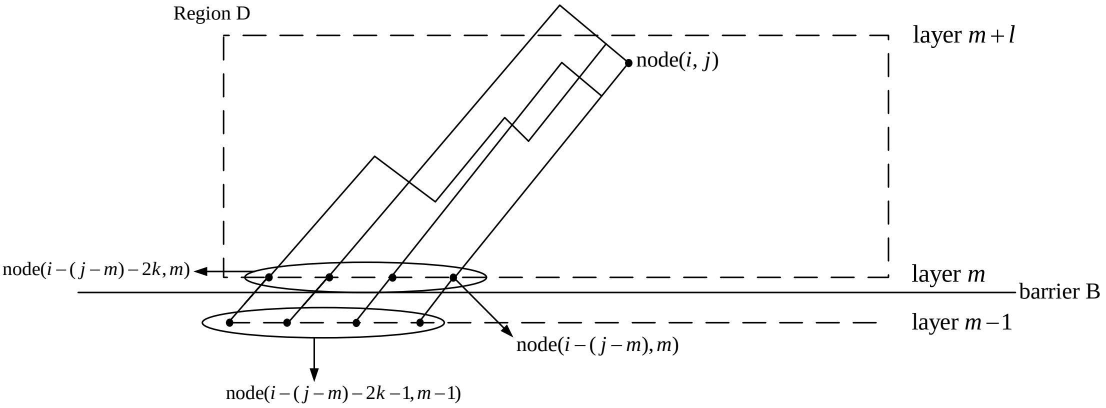
⊙ For any node(i, j) in this region, some still alive paths reaching its parent nodes have already stay continuously above the barrier for $l - 1$ periods. Since node(i, j) is above the barrier, if considering one more step to reach node(i, j) from its parent nodes, some paths may be knocked out when reaching node (i, j). Thus, it must be the case that $g(i, j) \leq g(i - 1, j + 1) + g(i - 1, j - 1)$. In other words, it is impossible to use the information of $g(i - 1, j + 1)$ and $g(i - 1, j - 1)$ of parent nodes to calculate $g(i, j)$. Take the node Z in Figure 8-13 as an example.

Figure 8-13 The numbers in the parentheses at each node is $(g(i,j), \bar{g}(i,j))$, where $\bar{g}(i,j)$ is the number of paths that the maximum sojourn time above B is greater than or equal to l.



⊙ The method based on the reflection principle to calculate $g(i,j)$:

Figure 8-14



The paths contribute to $g(i,j)$ must satisfy the following three conditions:

1. Consider the number of still alive paths for node$(i-(j-m)-2k-1, m-1)$, i.e., $g(i-(j-m)-2k-1, m-1)$. Since these nodes are below the barrier B, the still alive paths of these nodes do not accumulate any number of period sojourning above the barrier B.
2. Move upward one step to node$(i-(j-m)-2k, m)$, which is just above the barrier.
3. Start from node$(i-(j-m)-2k, m)$ to node(i,j).
   3.1. Do not cross the barrier B from above.
   3.2. The number of sojourn period is smaller than l.

(i) The method to choose k: Since the number of steps from node$(i-(j-m)-2k, m)$ to (i,j) must smaller than l.

$$\Rightarrow i-(i-(j-m)-2k) = (j-m)+2k < l$$
$$\Rightarrow k < \tfrac{l-(j-m)}{2}$$

(ii) The number of all paths from node$(i-(j-m)-2k, m)$ to node(i,j) is $\begin{pmatrix} j-m+2k \\ j-m+k \end{pmatrix}$

> Total number of steps is $(j-m)+2k$, and the number of net upward steps is $j-m$.
> $\Rightarrow$ The number of upward steps is $j-m+k$, and the number of downward steps is k.

(iii) The number of paths which are from node$(i-(j-m)-2k, m)$ to node(i,j) and not crossing B from above

= the results in (ii) − number of paths form node$(i-(j-m)-2k, m)$ to node(i,j) which touch or cross the layer $m-1$

= the results in (ii) − $\begin{pmatrix} j-m+2k \\ j-m+k+1 \end{pmatrix}$

> By the reflection principle, the second term in the above equation equals the number of paths from node$(i-(j-m)-2k, m-2)$ to node(i,j).
>
> Total number of steps is $j-m+2k$, and the number of net upward steps is $j-m+2$.
> $\Rightarrow$ The number of upward steps is $j-m+k+1$, and the number of downward steps is $k-1$.

⊙ For Parisian up-and-out call,

option value=$e^{-rT}\sum\limits_{a=0}^{n} g(n,j)p^a(1-p)^{n-a}\max(Su^a d^{n-a}-K,0)$, where $j=2a-n$.