CHAPTER SEVENTEEN

Trees

I love a tree more than a man.

Ludwig van Beethoven (1770–1827)

This chapter starts with combinatorial methods to speed up European option pricing. The influential and versatile trinomial model is also introduced. These tree are regarded as more "accurate" than binomial trees. Then an important maxim is brought up: The comparison of algorithms should be based on the actual running time. This chapter ends with multinomial trees for pricing multivariate derivatives.

17.1 Pricing Barrier Options with Combinatorial Methods

We first review the binomial approximation to the geometric Brownian motion $S = e^X$, where X is a $(\mu - \sigma^2/2, \sigma)$ Brownian motion. (Equivalently, $dS/S = \mu\, dt + \sigma\, dW$.) For economy of expression, we use S in place of $S(0)$ for the current time-zero price. Consider the stock price at time $\Delta t \equiv \tau/n$, where τ is the time to maturity. From Eq. (13.12),

$$E[\, S(\Delta t)\,] = Se^{\mu \Delta t}, \quad \mathrm{Var}[\, S(\Delta t)\,] = S^2 e^{2\mu \Delta t}\big(e^{\sigma^2 \Delta t} - 1\big) \to S^2 \sigma^2 \Delta t.$$

Under the binomial model, the stock price increases to Su with probability q or decreases to Sd with probability $1 - q$ at time Δt. The expected stock price at time Δt is $qSu + (1 - q)\, Sd$. Our first requirement is that it converge to $Se^{\mu \Delta t}$. The variance of the stock price at time Δt is given by $q(Su)^2 + (1 - q)(Sd)^2 - (Se^{\mu \Delta t})^2$. Our second requirement is that it converge to $S^2\sigma^2 \Delta t$. With $ud = 1$ imposed, the choice below works:

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}, \quad q = \frac{e^{\mu \Delta t} - d}{u - d}. \tag{17.1}$$

In a risk-neutral economy, $\mu = r$ and q approaches

$$p \equiv \frac{1}{2} + \frac{1}{2}\, \frac{r - \sigma^2/2}{\sigma} \sqrt{\Delta t}$$

by convergence (9.17). We set $\mu' \equiv r - \sigma^2/2$ throughout.

The combinatorial method is as elementary as it is elegant. It can often cut the running time by an order of magnitude. The basic paradigm is to count the number of admissible paths that lead from the root to any terminal node. We first used this method in the linear-time algorithm for standard European option pricing in Fig. 9.9, and now we apply it to barrier option pricing. The reflection principle provides the necessary tool.

17.1.1 The Reflection Principle

Imagine a particle at position $(0, -\boldsymbol{a})$ on the integral lattice that is to reach $(n, -\boldsymbol{b})$. Without loss of generality, assume that $\boldsymbol{a}, \boldsymbol{b} \geq 0$. This particle is constrained to move to $(i+1, j+1)$ or $(i+1, j-1)$ from (i, j), as shown below:

$$(i, j) \begin{cases} \nearrow (i+1, j+1) & \text{associated with the up move } S \to Su \\ \searrow (i+1, j-1) & \text{associated with the down move } S \to Sd \end{cases}.$$

How many paths touch the x axis?

For a path from $(0, -\boldsymbol{a})$ to $(n, -\boldsymbol{b})$ that touches the x axis, let J denote the first point at which this happens. When the portion of the path from $(0, -\boldsymbol{a})$ to J is reflected, a path from $(0, \boldsymbol{a})$ to $(n, -\boldsymbol{b})$ is constructed, which also hits the x axis at J for the first time (see Fig. 17.1). The one-to-one mapping shows that the number of paths from $(0, -\boldsymbol{a})$ to $(n, -\boldsymbol{b})$ that touch the x axis equals the number of paths from $(0, \boldsymbol{a})$ to $(n, -\boldsymbol{b})$. This is the celebrated **reflection principle** of André (1840–1917) published in 1887 [604, 686]. Because a path of this kind has $(n+\boldsymbol{b}+\boldsymbol{a})/2$ down moves and $(n-\boldsymbol{b}-\boldsymbol{a})/2$ up moves, there are

$$\binom{n}{\frac{n+\boldsymbol{a}+\boldsymbol{b}}{2}} \quad \text{for even } n+\boldsymbol{a}+\boldsymbol{b} \tag{17.2}$$

such paths. The convention here is $\binom{n}{k} = 0$ for $k < 0$ or $k > n$.

➤ **Exercise 17.1.1** What is the probability that the stock's maximum price is at least Su^k?

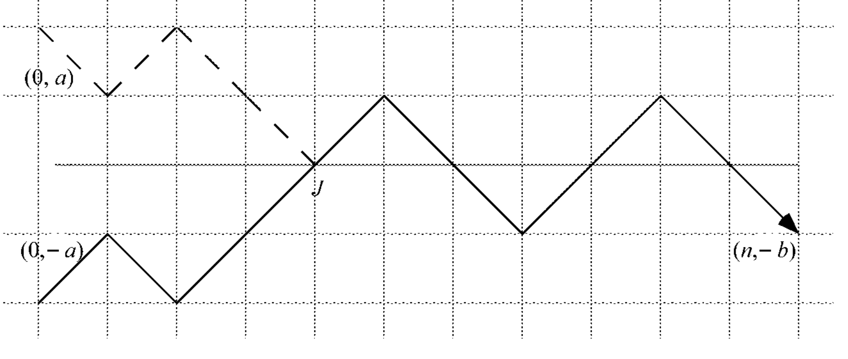

Figure 17.1: The reflection principle for binomial random walks. Two paths of equal length will be separated by a distance of $2k$ on the binomial tree if their respective accumulative numbers of up moves differ by k (see Eq. (17.4)).

17.1.2 Combinatorial Formulas for Barrier Options

We focus on the down-and-in call with barrier $H < X$. Assume that $H < S$ without loss of generality for, otherwise, the option is identical to a standard call. Define

$$a \equiv \left\lceil \frac{\ln(X/(Sd^n))}{\ln(u/d)} \right\rceil = \left\lceil \frac{\ln(X/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rceil,$$
$$h \equiv \left\lfloor \frac{\ln(H/(Sd^n))}{\ln(u/d)} \right\rfloor = \left\lfloor \frac{\ln(H/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rfloor. \tag{17.3}$$

Both a and h have straightforward interpretations. First, h is such that $\tilde{H} \equiv Su^h d^{n-h}$ is the terminal price that is closest to, but does not exceed, H. The true barrier is replaced with the **effective barrier** $\tilde{H}$ in the binomial model. Similarly, a is such that $\tilde{X} \equiv Su^a d^{n-a}$ is the terminal price that is closest to, but not exceeded by, X. A process with n moves hence ends up in the money if and only if the number of up moves is at least a.

The price $Su^k d^{n-k}$ is at a distance of $2k$ from the lowest possible price Sd^n on the binomial tree because

$$Su^k d^{n-k} = Sd^{-k} d^{n-k} = Sd^{n-2k}. \tag{17.4}$$

Given this observation, Fig. 17.2 plots the relative distances of various prices on the tree.

The number of paths from S to the terminal price $Su^j d^{n-j}$ is $\binom{n}{j}$, each with probability $p^j(1-p)^{n-j}$. With reference to Fig. 17.2, we can apply the reflection principle with $\boldsymbol{a} = n - 2h$ and $\boldsymbol{b} = 2j - 2h$ in formula (17.2) by treating the S line as the x axis. Therefore

$$\binom{n}{\frac{n+(n-2h)+(2j-2h)}{2}} = \binom{n}{n-2h+j}$$

paths hit $\tilde{H}$ in the process for $h \leq n/2$. We conclude that the terminal price $Su^j d^{n-j}$ is reached by a path that hits the effective barrier with probability

$$\binom{n}{n-2h+j} p^j(1-p)^{n-j}, \tag{17.5}$$

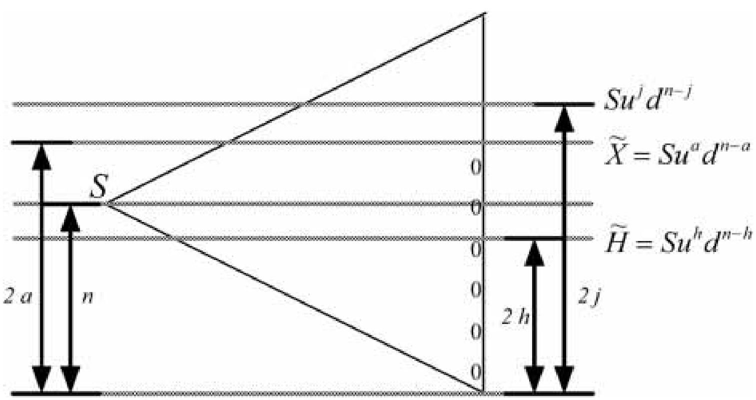

Figure 17.2: Down-and-in call and binomial tree. The effective barrier is the $\tilde{H}$ line, and the process starts on the S line.

Linear-time, constant-space algorithm for pricing down-and-in calls on a non-dividend-paying stock:

```
input:   S, σ, X, H (H < X, H < S), n, τ, r;
real     p, u, d, b, D, C;
integer  j, a, h;
u := e^{σ√(τ/n)}; d := e^{−σ√(τ/n)};
a := ⌈ln(X/Sd^n)/ln(u/d)⌉; h := ⌊ln(H/Sd^n)/ln(u/d)⌋;
p := (e^{rτ/n} − d)/(u − d);   // Risk-neutral probability.
b := p^{2h}(1 − p)^{n−2h};   // b_{2h} is computed.
D := S × u^{2h} d^{n−2h}; C := b × (D − X);
for (j = 2h − 1 down to a) {
        b := b × p × (n − 2h + j + 1)/((1 − p) × (2h − j));
        D := D × u/d;
        C := C + b × (D − X);
}
return C/e^{rτ};
```

Figure 17.3: Optimal algorithm for European down-and-in calls on a stock that does not pay dividends. Variable b stores $b_j \equiv \binom{n}{n-2h+j} p^j (1-p)^{n-j}$ for $j = 2h, 2h-1, \ldots, a$, in that order, and variable C accumulates the summands in option value (17.6) for $j = 2h, 2h-1, \ldots, a$. Note that $b_j = b_{j+1}\,[\,(1-p)(n-2h+j+1)\,]/[\,p(2h-j)\,]$. The structure is similar to the one in Fig. 9.9.

and the option value equals

$$R^{-n} \sum_{j=a}^{2h} \binom{n}{n-2h+j} p^j (1-p)^{n-j} \left(S u^j d^{n-j} - X \right), \tag{17.6}$$

where $R \equiv e^{r\tau/n}$ is the riskless return per period. Formula (17.6) is an alternative characterization of the binomial tree algorithm [624]. It also implies a linear-time algorithm (see Fig. 17.3). In fact, the running time is proportional to $2h - a$, which is close to $n/2$:

$$2h - a \approx \frac{n}{2} + \frac{\ln(H^2/(SX))}{2\sigma\sqrt{\tau/n}} = \frac{n}{2} + O(\sqrt{n}).$$

The preceding methodology has applications to exotic options whose terminal payoff is "nonstandard" and closed-form solutions are hard to come by. Discrete-time models may also be more realistic than continuous-time ones for contracts based on discrete sampling of the price process at regular time intervals [147, 597].

EXAMPLE 17.1.1 A binary call pays off \$1 if the underlying asset finishes above the strike price and nothing otherwise. The price of a binary down-and-in call is formula (17.6) with $Su^j d^{n-j} - X$ replaced with 1.

EXAMPLE 17.1.2 A **power option** pays off $\max([\,S(\tau) - X\,]^p, 0)$ (sometimes $\max(S(\tau)^p - X, 0)$) at expiration [894]. To price a down-and-in power option, replace $Su^j d^{n-j} - X$ in formula (17.6) with $(Su^j d^{n-j} - X)^p$ $((Su^j d^{n-j})^p - X$, respectively).

➤ **Exercise 17.1.2** Derive pricing formulas similar to formula (17.6) for the other three barrier options: down-and-out, up-and-in, and up-and-out options.

➤ **Exercise 17.1.3** Use the reflection principle to derive a combinatorial pricing formula for the European lookback call on the minimum.

➤ **Exercise 17.1.4** Consider the **exploding call spread**, which has the same payoff as the bull call spread except that it is exercised promptly the moment the stock price touches the trigger price K [377]. (1) Write a combinatorial formula for the value of this path-dependent option. (2) Verify that the valuation of the option runs in linear time.

➤ **Exercise 17.1.5** Derive a combinatorial pricing formula for the reset call option.

➤ **Exercise 17.1.6** Prove that option value (17.6) converges to value (11.4) with $q = 0$.

➤ **Exercise 17.1.7** Derive a pricing formula for the European power option $\max(S(\tau)^2 - X, 0)$.

➢ **Programming Assignment 17.1.8** Design fast algorithms for European barrier options.

➢ **Programming Assignment 17.1.9** Implement $O(n^3)$-time algorithms for European geometric average-rate options with combinatorics, improving Programming Assignment 11.7.6.

➢ **Programming Assignment 17.1.10** (1) Implement $O(n^2)$-time algorithms for European lookback options, improving Programming Assignment 11.7.11, part (1). (2) Improve the running time to $O(n)$.

17.1.3 Convergence of Binomial Tree Algorithms

Option value (17.6) results in the sawtoothlike convergence shown in Fig. 11.5. Increasing n therefore does not necessarily lead to more accurate results. The reasons are not hard to see. The true barrier most likely does not equal the effective barrier. The same holds for the strike price and the effective strike price. Both introduce specification errors [271]. The issue of the strike price is less critical as evinced by the fast convergence of binomial tree algorithms for standard European options. The issue of the barrier is not negligible, however, because the barrier exerts its influence throughout the price dynamics.

Figure 17.4 suggests that convergence is actually good if we limit n to certain values – 191 in the figure, for example. These values make the true barrier coincide with or occur just above one of the stock price levels, that is, $H \approx Sd^j = Se^{-j\sigma\sqrt{\tau/n}}$ for some integer j [111, 196]. The preferred n's are thus

$$n = \left\lfloor \frac{\tau}{[\,\ln(S/H)/(j\sigma)\,]^2} \right\rfloor, \quad j = 1, 2, 3, \ldots$$

There is only one minor technicality left. We picked the effective barrier to be one of the $n+1$ possible terminal stock prices. However, the effective barrier above, Sd^j, corresponds to a terminal stock price only when $n - j$ is even by Eq. (17.4).* To close this gap, we decrement n by one, if necessary, to make $n - j$ an even number.

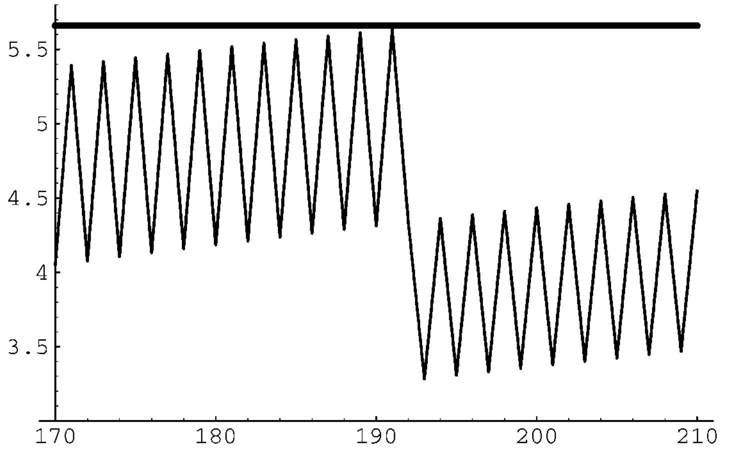

Figure 17.4: Convergence of binomial model for down-and-in calls. A detailed look of Fig. 11.5. Note that the approximation is quite close (5.63542 vs. the analytical value 5.6605) at $n = 191$.

The preferred n's are now

$$n = \begin{cases} \ell, & \text{if } \ell - j \text{ is even} \\ \ell - 1, & \text{otherwise} \end{cases}, \quad \ell \equiv \left\lfloor \frac{\tau}{[\,\ln(S/H)/(j\sigma)\,]^2} \right\rfloor \tag{17.7}$$

$j = 1, 2, 3, \ldots$. In summary, evaluate pricing formula (17.6) only with the n's above. The result is shown in Fig. 17.5.

Now that barrier options can be efficiently priced, we can afford to pick very large n's. This has profound consequences. For example, pricing seems prohibitively time consuming when $S \approx H$ because n, being proportional to $1/\ln^2(S/H)$, is large. This observation is indeed true of standard quadratic-time binomial tree algorithms like the one in Fig. 11.4. However, it no longer applies to the linear-time algorithm [624].

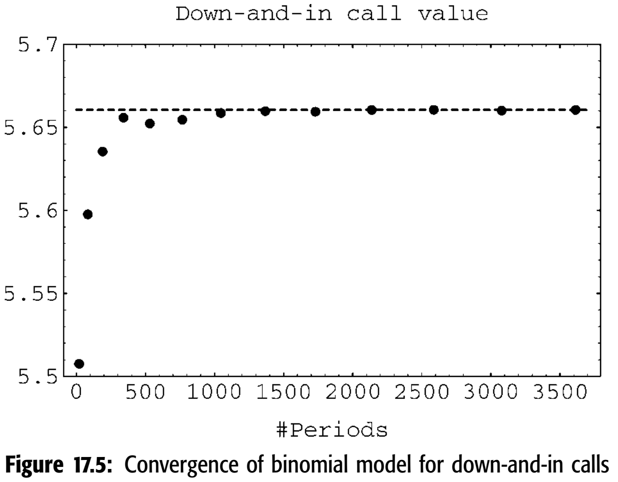

Figure 17.5: Convergence of binomial model for down-and-in calls at well-chosen n's. Formula (17.6) is evaluated at $n = 21$ (1), 84 (2), 191 (3), 342 (4), 533 (5), 768 (6), 1047 (7), 1368 (8), 1731 (9), 2138 (10), 2587 (11), 3078 (12), and 3613 (13), with the corresponding j in parentheses.

➤ **Exercise 17.1.11** How do we efficiently price a portfolio of barrier options with identical underlying assets but different barriers under the binomial model?

➤ **Exercise 17.1.12** Explain why Fig. 11.5 shows that the calculated values underestimate the analytical value.

➤ **Exercise 17.1.13** In formula (17.6), the barrier H is replaced with the effective barrier $\tilde{H}$, which is one of the $n+1$ terminal prices. If the effective barrier is allowed to be one of all possible $2n+1$ prices $Su^n, Su^{n-1}, \ldots, Su^{-n}$, what changes should be made to formulas (17.6) and (17.7)?

➤ **Programming Assignment 17.1.4** Try $\ell \equiv \lceil \frac{\tau}{[\ln(S/H)/(j\sigma)]^2} \rceil$ instead.

17.1.4 Double-Barrier Options

Double-barrier options contain two barriers, L and H, with $L < H$. Depending on how the barriers affect the security, various barrier options can be defined. We consider options that come into existence if and only if *either* barrier is hit.

A particle starts at position $(0, -\boldsymbol{a})$ on the integral lattice and is destined for $(n, -\boldsymbol{b})$. Without loss of generality, assume that $\boldsymbol{a}, \boldsymbol{b} \geq 0$. The number of paths in which a hit of the H line, $x = 0$, appears before a hit of the L line $x = -s$ is

$$\binom{n}{\frac{n+\boldsymbol{a}-\boldsymbol{b}+2s}{2}} \quad \text{for even } n+\boldsymbol{a}-\boldsymbol{b}. \tag{17.8}$$

In the preceding expression, we assume that $s > \boldsymbol{b}$ and $\boldsymbol{a} < s$ to make both barriers effective.

The preceding expression can be generalized. Let A_i denote the set of paths that hit the barriers with a hit sequence *containing* $\overbrace{H^+ L^+ H^+ \cdots}^{i}$, $i \geq 2$, where L^+ denotes a sequence of Ls and H^+ denotes a sequence of Hs. Similarly, let B_i denote the set of paths that hit the barriers with a sequence *containing* $\overbrace{L^+ H^+ L^+ \cdots}^{i}$, $i \geq 2$. For instance, a path with the hit pattern $LLHLLHH$ belongs to A_2, A_3, B_2, B_3, and B_4. Note that $A_i \cap B_i$ may not be empty. The number of paths that hit either barrier is

$$\boldsymbol{N}(\boldsymbol{a}, \boldsymbol{b}, s) = \sum_{i=1}^{n} (-1)^{i-1} (|A_i| + |B_i|). \tag{17.9}$$

The calculation of summation (17.9) can stop at the first i when $|A_i| + |B_i| = 0$.

The value of the double-barrier call is now within reach. Let us take care of the degenerate cases first. If $S \leq L$, the double-barrier call is reduced to a standard call. If $S \geq H$, it is reduced to a knock-in call with a single barrier H. So we assume that $L < S < H$ from now on. Under this assumption, it is easy to check that the double-barrier option is reduced to simpler options unless $L < X < H$. So we assume that $L < X < H$ from now on. Define

$$h \equiv \left\lceil \frac{\ln(H/(Sd^n))}{\ln(u/d)} \right\rceil = \left\lceil \frac{\ln(H/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rceil,$$

$$l \equiv \left\lfloor \frac{\ln(L/(Sd^n))}{\ln(u/d)} \right\rfloor = \left\lfloor \frac{\ln(L/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rfloor.$$

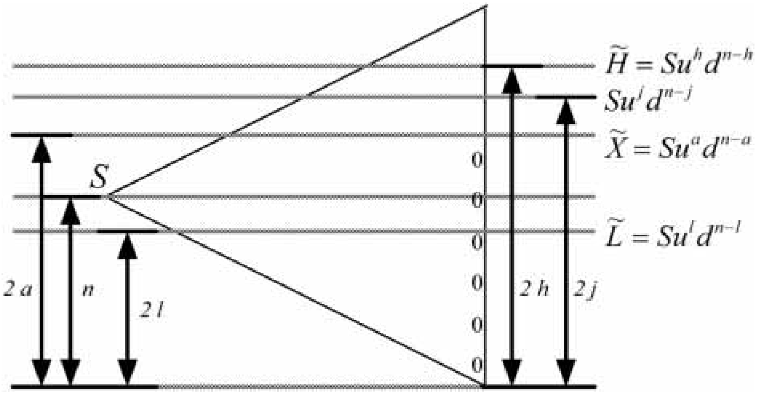

Figure 17.6: Double-barrier call under binomial model. The effective barriers are the $\tilde{H}$ line and the $\tilde{L}$ line, and the process starts on the S line.

The barriers will be replaced with the effective barriers $\tilde{H} \equiv Su^h d^{n-h}$ and $\tilde{L} \equiv Su^l d^{n-l}$. Note that in Eq. (17.9), only terminal nodes between $\tilde{L}$ and $\tilde{H}$ (inclusive) are considered. These terminal nodes together contribute

$$A \equiv R^{-n} \sum_{j=a}^{h} \boldsymbol{N}(2h-n, 2h-2j, 2(h-l))\, p^j (1-p)^{n-j} \left(Su^j d^{n-j} - X\right) \tag{17.10}$$

to the option value, where a is defined in Eqs. (17.3). See Fig. 17.6 for the relative positions of of the various parameters. As for the terminal nodes outside the above-mentioned range, they constitute a standard call with a strike price of $\tilde{H}u^2$. Let its value be D. The double-barrier call thus has value $A + D$. The convergence is sawtoothlike [179].

➤ **Exercise 17.1.15** Prove formula (17.8).

➤ **Exercise 17.1.16** Apply the reflection principle repetitively to verify that

$$|A_i| = \begin{cases} \binom{n}{\frac{n+\boldsymbol{a}+\boldsymbol{b}+(i-1)s}{2}} & \text{for odd } i \\ \binom{n}{\frac{n+\boldsymbol{a}-\boldsymbol{b}+is}{2}} & \text{for even } i \end{cases}, \quad |B_i| = \begin{cases} \binom{n}{\frac{n-\boldsymbol{a}-\boldsymbol{b}+(i+1)s}{2}} & \text{for odd } i \\ \binom{n}{\frac{n-\boldsymbol{a}+\boldsymbol{b}+is}{2}} & \text{for even } i \end{cases}.$$

Assume that $n + \boldsymbol{a} - \boldsymbol{b}$ is even for $|A_i|$ and $n - \boldsymbol{a} + \boldsymbol{b}$ is even for $|B_i|$.

➤ **Exercise 17.1.17** Prove Eq. (17.9).

➤ **Exercise 17.1.18** (1) Formulate the in–out parity for double-barrier options. (2) Replicate the double-barrier option by using knock-in options, knock-out options, and double-barrier options that come into existence if and only if both barriers are hit. (3) Modify Eq. (17.9) to price the double-barrier option in (2).

➤ **Exercise 17.1.19** Consider a generalized double-barrier option with the two barriers defined by functions f_ℓ and f_h, where $f_\ell(t) < f_h(t)$ for $t \geq 0$. Transform it into a double-barrier option with constant barriers by changing the underlying price process.

https://doi.org/10.1017/CBO9780511546839.018 Published online by Cambridge University Press

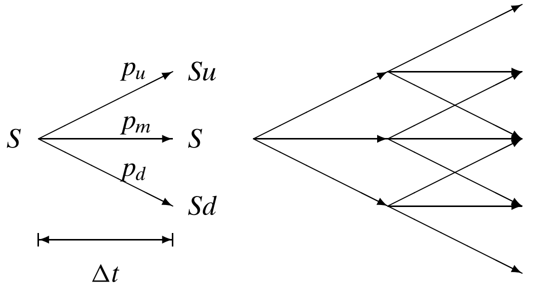

Figure 17.7: Trinomial model. There are three branches from each node.

➢ **Programming Assignment 17.1.20** Implement a linear-time algorithm for double-barrier options.

17.2 Trinomial Tree Algorithms

We now set up the trinomial approximation to the geometric Brownian motion $dS/S = r\,dt + \sigma\,dW$ [107]. The three stock prices at time Δt are S, Su, and Sd, where $ud = 1$ (see Fig. 17.7). Impose the matching of mean and that of variance to obtain

$$\begin{aligned} 1 &= p_u + p_m + p_d, \\ SM &\equiv [\, p_u u + p_m + (p_d/u)\,]\, S, \\ S^2 V &\equiv p_u (Su - SM)^2 + p_m (S - SM)^2 + p_d (Sd - SM)^2, \end{aligned}$$

where $M \equiv e^{r\Delta t}$ and $V \equiv M^2(e^{\sigma^2 \Delta t} - 1)$ by Eqs. (6.11). It is easy to verify that

$$\begin{aligned} p_u &= \frac{u(V + M^2 - M) - (M-1)}{(u-1)(u^2-1)}, \\ p_d &= \frac{u^2(V + M^2 - M) - u^3(M-1)}{(u-1)(u^2-1)}. \end{aligned}$$

We need to make sure that the probabilities lie between zero and one. Use $u = e^{\lambda\sigma\sqrt{\Delta t}}$, where $\lambda \geq 1$ is a parameter that can be tuned. Then

$$p_u \to \frac{1}{2\lambda^2} + \frac{(r + \sigma^2)\sqrt{\Delta t}}{2\lambda\sigma}, \quad p_d \to \frac{1}{2\lambda^2} - \frac{(r - 2\sigma^2)\sqrt{\Delta t}}{2\lambda\sigma}.$$

A nice choice for λ is $\sqrt{\pi/2}$ [824].

➤ **Exercise 17.2.1** Verify the following: (1) $\ln(S(\Delta t)/S)$ has mean $\mu' \Delta t$, (2) the variance of $\ln(S(\Delta t)/S)$ converges to $\sigma^2 \Delta t$, and (3) $S(\Delta t)$'s mean converges to $Se^{r\Delta t}$.

➤ **Exercise 17.2.2** The trinomial model no longer supports perfect replication of options with stocks and bonds as in the binomial model. Replicating an option with

h shares of stock and \$$B$ in bonds involves two unknowns h and B, but the three branches imply three conditions. Give an example for which resulting system of three equations in two knowns is inconsistent.

➢ **Programming Assignment 17.2.3** Recall the diagonal method in Section 9.7. Write a program to perform backward induction on the trinomial tree with the diagonal method.

17.2.1 Pricing Barrier Options

Binomial tree algorithms introduce a specification error by replacing the barrier with a nonidentical effective barrier. The trinomial tree algorithm that is due to Ritchken solves the problem by adjusting λ so that the barrier is hit exactly [745]. Here is the idea. It takes

$$h = \frac{\ln(S/H)}{\lambda\sigma\sqrt{\Delta t}}$$

consecutive down moves to go from S to H if h is an integer, which is easy to achieve by adjusting λ. Typically, we find the smallest $\lambda \geq 1$ such that h is an integer, that is,

$$\lambda = \min_{j=1,2,3,\ldots} \frac{\ln(S/H)}{j\sigma\sqrt{\Delta t}}.$$

This done, one of the layers of the trinomial tree coincides with the barrier. We note that such a λ may not exist for very small n's. The following probabilities may be used:

$$p_u = \frac{1}{2\lambda^2} + \frac{\mu'\sqrt{\Delta t}}{2\lambda\sigma}, \quad p_m = 1 - \frac{1}{\lambda^2}, \quad p_d = \frac{1}{2\lambda^2} - \frac{\mu'\sqrt{\Delta t}}{2\lambda\sigma}.$$

Note that this particular trinomial model reduces to the binomial model when $\lambda = 1$. See Fig. 17.8 for the algorithm. Figure 17.9 shows the trinomial model's convergence behavior. If the stock pays a continuous dividend yield of q, then we let $\mu' \equiv r - q - \sigma^2/2$.

➤ **Exercise 17.2.4** It was shown in Subsection 10.2.2 that binomial trees can be extended backward in time for two periods to compute delta and gamma. Argue that trinomial trees need to be extended backward in time for only one period to compute the same hedge parameters.

➤ **Exercise 17.2.5** Derive combinatorial formulas for European down-and-in, down-and-out, up-and-in, and up-and-out options.

➢ **Programming Assignment 17.2.6** Implement trinomial tree algorithms for barrier options. Add rebates for the knock-out type.

17.2.2 Remarks on Algorithm Comparison

Algorithms are often compared based on the n value at which they converge, and the one with the smallest n wins. This is a fallacy as it implies that giraffes are faster than cheetahs simply because they take fewer strides to travel the same distance,

Trinomial tree algorithm for down-and-out calls on a non-dividend-paying stock:

```
input:   S, σ, X, H (H < X, H < S), n, τ, r;
real     u, d, p_u, p_m, p_d, λ, Δt, C[2n+1];
integer i, j, h;
Δt := τ/n;
h =: ⌊ln(S/H)/(σ√Δt)⌋;
if [ h < 1 or h > n ] return failure;
λ := ln(S/H)/(hσ√Δt);
p_u := 1/(2λ^2) + (r − σ^2/2)√Δt/(2λσ);
p_d := 1/(2λ^2) − (r − σ^2/2)√Δt/(2λσ);
p_m := 1 − p_u − p_d;
u := e^(λσ√Δt);
for (i = 0 to 2n) { C[i] := max(0, Su^(n−i) − X); }
C[n+h] := 0;   // A hit.
for (j = n−1 down to 0) {
        for (i = 0 to 2j)
          C[i] := p_u C[i] + p_m C[i+1] + p_d C[i+2];
        if [ j + h ≤ 2j ] C[j+h] := 0;   // A hit.
}
return C[0]/e^(rτ);
```

Figure 17.8: Trinomial tree algorithm for down-and-out calls on a non-dividend-paying stock. The barrier $H = Su^{-h}$ corresponds to $C[h + j]$ at times $j = n, n-1, \ldots, h$. It is not hard to show that h must be at least $\sigma^2\tau/\ln^2(S/H)$ to make $\lambda \geq 1$. This algorithm should be compared with the ones in Figs. 11.4 and 33.2.

forgetting that how fast the legs move is equally critical. like any race, an algorithm's performance must be based on its actual running time [717]. As a concrete example, Figs. 11.5 and 17.9 show that the trinomial model converges at a smaller n than the binomial model. It is in this sense when people say that trinomial models

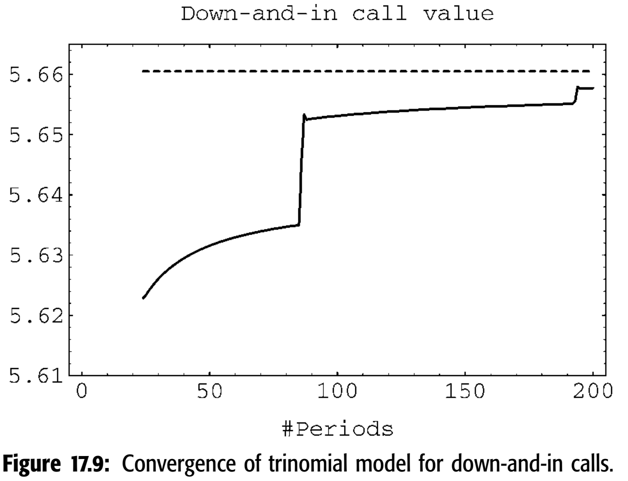

Figure 17.9: Convergence of trinomial model for down-and-in calls. Plotted are the down-and-in call values as computed by the trinomial tree algorithm against the number of time periods. The parameters are identical to those used in Fig. 11.5. The analytical value 5.6605 is plotted for reference.

converge faster than binomial ones. However, the linear-time binomial tree algorithm for European barrier options actually performs better than the trinomial counterpart [610, 624].

17.3 Pricing Multivariate Contingent Claims

Multivariate derivatives such as correlation options are contingent claims that depend on two or more underlying assets. Consider the **basket option** on m assets. The basket call has the terminal payoff $\max(\sum_{i=1}^{m} \alpha_i S_i(\tau) - X, 0)$, whereas the basket put has the terminal payoff $\max(X - \sum_{i=1}^{m} \alpha_i S_i(\tau), 0)$, where α_i is the percentage of asset i [663]. Basket options are essentially options on a portfolio of stocks or index options on a capitalization- or a price-weighted index. Consider the **option on the best of two risky assets and cash** as another example. It has a terminal payoff of $\max(S_1(\tau), S_2(\tau), X)$, which guarantees a cash flow of X and the better of two assets, say a stock fund and a bond fund [833]. Because the terminal payoff can be written as $X + \max(\max(S_1(\tau), S_2(\tau)) - X, 0)$, the option is worth $Xe^{-r\tau} + C$, where C is the price of a call option on the maximum of two assets with strike price X. This section presents binomial and trinomial models for multiple underlying assets to price multivariate derivatives [107, 110]. The aim is to construct a multivariate discrete-time probability distribution with the desired means and variance–covariance values.

17.3.1 Construction of a Correlated Trinomial Model

Suppose that two risky assets S_1 and S_2 follow $dS_i/S_i = r\,dt + \sigma_i\,dW_i$ in a risk-neutral economy, $i = 1, 2$. Define $M_i \equiv e^{r\Delta t}$ and $V_i \equiv M_i^2(e^{\sigma_i^2 \Delta t} - 1)$, where $S_i M_i$ is the mean and $S_i^2 V_i$ is the variance of S_i at time Δt from now. The value of $S_1 S_2$ at time Δt has a joint lognormal distribution with mean $S_1 S_2 M_1 M_2 e^{\rho\sigma_1\sigma_2\Delta t}$, where ρ is the correlation between dW_1 and dW_2. We proceed to match the first and the second moments of the approximating discrete distribution to those of the continuous counterpart. At time Δt from now, there are five distinct outcomes. The five-point probability distribution of the asset prices is (as usual, we impose $u_i d_i = 1$)

Probability	*Asset 1*	*Asset 2*
p_1	$S_1 u_1$	$S_2 u_2$
p_2	$S_1 u_1$	$S_2 d_2$
p_3	$S_1 d_1$	$S_2 d_2$
p_4	$S_1 d_1$	$S_2 u_2$
p_5	S_1	S_2

The probabilities must sum to one, and the means must be matched, leading to

$$\begin{aligned} 1 &= p_1 + p_2 + p_3 + p_4 + p_5, \\ S_1 M_1 &= (p_1 + p_2)\, S_1 u_1 + p_5 S_1 + (p_3 + p_4)\, S_1 d_1, \\ S_2 M_2 &= (p_1 + p_4)\, S_2 u_2 + p_5 S_2 + (p_2 + p_3)\, S_2 d_2. \end{aligned}$$

The following equations match the variances and the covariance:

$$\begin{aligned}
S_1^2 V_1 &= (p_1+p_2)[\,(S_1u_1)^2-(S_1M_1)^2\,]+p_5[\,S_1^2-(S_1M_1)^2\,]\\
&\quad +(p_3+p_4)[\,(S_1d_1)^2-(S_1M_1)^2\,],\\
S_2^2 V_2 &= (p_1+p_4)[\,(S_2u_2)^2-(S_2M_2)^2\,]+p_5[\,S_2^2-(S_2M_2)^2\,]\\
&\quad +(p_2+p_3)[\,(S_2d_2)^2-(S_2M_2)^2\,],\\
S_1S_2R &= (p_1u_1u_2+p_2u_1d_2+p_3d_1d_2+p_4d_1u_2+p_5)\,S_1S_2,
\end{aligned}$$

where $R \equiv M_1M_2e^{\rho\sigma_1\sigma_2\Delta t}$. The solutions are

$$\begin{aligned}
p_1 &= \frac{u_1u_2(R-1)-f_1(u_1^2-1)-f_2(u_2^2-1)+(f_2+g_2)(u_1u_2-1)}{(u_1^2-1)\,(u_2^2-1)},\\
p_2 &= \frac{-u_1u_2(R-1)+f_1(u_1^2-1)\,u_2^2+f_2(u_2^2-1)-(f_2+g_2)(u_1u_2-1)}{(u_1^2-1)\,(u_2^2-1)},\\
p_3 &= \frac{u_1u_2(R-1)-f_1(u_1^2-1)\,u_2^2+g_2(u_2^2-1)\,u_1^2+(f_2+g_2)\,(u_1u_2-u_2^2)}{(u_1^2-1)\,(u_2^2-1)},\\
p_4 &= \frac{-u_1u_2(R-1)+f_1(u_1^2-1)+f_2(u_2^2-1)\,u_1^2-(f_2+g_2)(u_1u_2-1)}{(u_1^2-1)\,(u_2^2-1)},
\end{aligned}$$

where

$$\begin{aligned}
f_1 &= p_1+p_2=\frac{u_1(V_1+M_1^2-M_1)-(M_1-1)}{(u_1-1)\,(u_1^2-1)},\\
f_2 &= p_1+p_4=\frac{u_2(V_2+M_2^2-M_2)-(M_2-1)}{(u_2-1)\,(u_2^2-1)},\\
g_1 &= p_3+p_4=\frac{u_1^2(V_1+M_1^2-M_1)-u_1^3(M_1-1)}{(u_1-1)\,(u_1^2-1)},\\
g_2 &= p_2+p_3=\frac{u_2^2(V_2+M_2^2-M_2)-u_2^3(M_2-1)}{(u_2-1)\,(u_2^2-1)}.
\end{aligned}$$

Because $f_1+g_1=f_2+g_2$, we can solve for u_2 given $u_1=e^{\lambda\sigma_1\sqrt{\Delta t}}$ for an appropriate $\lambda>0$.

Once the tree is in place, a multivariate derivative can be valued by backward induction. The expected terminal value should be discounted at the riskless rate.

➤ **Exercise 17.3.1** Show that there are $1+2n(n+1)$ pairs of possible asset prices after n periods.

17.3.2 The Binomial Alternative

In the binomial model for m assets, asset i's price S_i can in one period go up to S_iu_i or down to S_id_i. There are thus 2^m distinct states after one step. (This illustrates the **curse of dimensionality** because the complexity grows exponentially in the dimension m.) We fix $u_i=e^{\sigma_i\sqrt{\Delta t}}$ and $u_id_i=1$. As working with the log price $\ln S_i$ turns out to be easier, we let $R_i\equiv\ln S_i(\Delta t)/S_i$. From Subsection 14.4.3, we know that $R_i\sim N(\mu_i'\Delta t,\sigma_i^2\Delta t)$, where $\mu_i'\equiv r-\sigma_i^2/2$.

We solve the $m = 2$ case first. Because (R_1, R_2) has a bivariate distribution, its moment generating function is

$$E\left[e^{t_1 R_1 + t_2 R_2} \right] = \exp\left[(t_1 \mu_1' + t_2 \mu_2')\, \Delta t + \left(t_1^2 \sigma_1^2 + t_2^2 \sigma_2^2 + 2 t_1 t_2 \sigma_1 \sigma_2 \rho \right) \frac{\Delta t}{2} \right]$$
$$\approx 1 + (t_1 \mu_1' + t_2 \mu_2')\, \Delta t + \left(t_1^2 \sigma_1^2 + t_2^2 \sigma_2^2 + 2 t_1 t_2 \sigma_1 \sigma_2 \rho \right) \frac{\Delta t}{2}.$$

Define the probabilities for up and down moves in the following table:

Probability	*Asset 1*	*Asset 2*
p_1	up	up
p_2	up	down
p_3	down	up
p_4	down	down

Under the binomial model, (R_1, R_2)'s moment generating function is

$$\begin{aligned} E\left[e^{t_1 R_1 + t_2 R_2} \right] &= p_1 e^{(t_1 \sigma_1 + t_2 \sigma_2)\sqrt{\Delta t}} + p_2 e^{(t_1 \sigma_1 - t_2 \sigma_2)\sqrt{\Delta t}} \\ &\quad + p_3 e^{(-t_1 \sigma_1 + t_2 \sigma_2)\sqrt{\Delta t}} + p_4 e^{(-t_1 \sigma_1 - t_2 \sigma_2)\sqrt{\Delta t}} \\ &\approx (p_1 + p_2 + p_3 + p_4) + t_1 \sigma_1 (p_1 + p_2 - p_3 - p_4) \sqrt{\Delta t} \\ &\quad + t_2 \sigma_2 (p_1 - p_2 + p_3 - p_4) \sqrt{\Delta t} \\ &\quad + \left[t_1^2 \sigma_1^2 + t_2^2 \sigma_2^2 + 2 t_1 t_2 \sigma_1 \sigma_2 (p_1 - p_2 - p_3 + p_4) \right] \frac{\Delta t}{2} \end{aligned}$$

Match the preceding two equations to obtain

$$p_1 = \frac{1}{4}\left[1 + \rho + \sqrt{\Delta t} \left(\frac{\mu_1'}{\sigma_1} + \frac{\mu_2'}{\sigma_2} \right) \right], \quad p_2 = \frac{1}{4}\left[1 - \rho + \sqrt{\Delta t} \left(\frac{\mu_1'}{\sigma_1} - \frac{\mu_2'}{\sigma_2} \right) \right],$$
$$p_3 = \frac{1}{4}\left[1 - \rho + \sqrt{\Delta t} \left(-\frac{\mu_1'}{\sigma_1} + \frac{\mu_2'}{\sigma_2} \right) \right], \quad p_4 = \frac{1}{4}\left[1 + \rho + \sqrt{\Delta t} \left(-\frac{\mu_1'}{\sigma_1} - \frac{\mu_2'}{\sigma_2} \right) \right].$$

Note its similarity to univariate case (9.17).

For the general case, we simply present the result as the methodology is identical. Let $\delta_i(j) = 1$ if S_i in state j makes an up move and -1 otherwise. Let $\delta_{ik}(j) = 1$ if S_i and S_k in state j make a move in the same direction and -1 otherwise. Then the probability that state j is reached is

$$p_j = \frac{1}{2^m} \sum_{\substack{i,k=1 \\ i<k}}^{m} \delta_{ik}(j)\, \rho_{ik} + \frac{1}{2^m} \sqrt{\Delta t} \sum_{i=1}^{m} \delta_i(j)\, \frac{\mu_i'}{\sigma_i}, \qquad j = 1, 2, \ldots, 2^m,$$

where ρ_{ik} denotes the correlation between dW_i and dW_k.

➤ **Exercise 17.3.2** It is easy to check that $\rho = 2(p_1 + p_4) - 1$. Show that this identity holds for any correlated binomial random walk defined by $R_1(i+1) - R_1(i) = \mu_1 \pm \sigma_1$ and $R_2(i+1) - R_2(i) = \mu_2 \pm \sigma_2$, where ρ denotes the correlation between R_1 and R_2 and "$\pm$" means "+" or "−" each with a probability of one half.

➤ **Exercise 17.3.3** With m assets, how many nodes does the tree have after n periods?

➤ **Exercise 17.3.4** (1) Write the combinatorial formula for a European call with terminal payoff $\max(S_1 S_2 - X, 0)$. (2) How fast can it be priced?

➤ **Exercise 17.3.5 (Optimal Hedge Ratio)** Derive the optimal number of futures to short in terms of minimum variance to hedge a long stock when the two assets are not perfectly correlated. Assume the horizon is Δt from now.

➢ **Programming Assignment 17.3.6** Implement the binomial model for the option on the best of two risky assets and cash.

Additional Reading

The combinatorial methods are emphasized in [342, 838]. Major ideas in lattice combinatorics can be found in [396, 675, 686]. See [434, 696, 835] for a more detailed analysis of the BOPM. An alternative approach to the convergence problem of binomial barrier option pricing is interpolation [271]. One more idea for tackling the difficulties in pricing various kinds of barrier options is to apply trinomial trees with varying densities [9]. More research on barrier options is discussed in [131, 158, 271, 370, 443, 444, 740, 752, 824, 841, 874]. See [68, 380, 423, 756] for numerical solutions of double-barrier options. The trinomial model is due to Parkinson [713]. The trinomial models presented here are improved in [202]. Consult [423, 519] and [114] for exact and approximation formulas for options on the maximum and the minimum of two assets, respectively. Further information on multivariate derivatives pricing can be found in [450, 451, 452, 539, 628, 876].

NOTE

1. We could have adopted the finer choice of the form Sd^j $(-n \leq j \leq n)$ for the effective barrier as the algorithm in Fig. 11.4 (see Exercise 17.1.13). This was not done in order to maintain similarity to binomial option pricing formula (9.10).