CHAPTER TWENTY-THREE

Introduction to Term Structure Modeling

How much of the structure of our theories really tells us about things in nature, and how much do we contribute ourselves?

Arthur Eddington (1882–1944)

The high interest rate volatility, especially since October 6, 1979 [401], calls for stochastic interest rate models. Models are also needed in managing interest rate risks of securities with interest-rate-sensitive cash flows. This chapter investigates stochastic term structure modeling with the **binomial interest rate tree** [779]. Simple as the model is, it illustrates most of the basic ideas underlying the models to come. The applications are also generic in that the pricing and hedging methodologies can be easily adapted to other models. Although the idea is similar to the one previously used in option pricing, the current task is complicated by two facts. First, the evolution of an entire term structure, not just a single stock price, is to be modeled. Second, interest rates of various maturities cannot evolve arbitrarily or arbitrage profits may result. The multitude of interest rate models is in sharp contrast to the single dominating model of Black and Scholes in option pricing.

23.1 Introduction

A stochastic interest rate model performs two tasks. First, it provides a stochastic process that defines future term structures. The ensuing dynamics must also disallow arbitrage profits. Second, the model should be "consistent" with the observed term structure [457]. Merton's work in 1970 marked the starting point of the continuous-time methodology to term structure modeling [493, 660]. This stochastic approach complements traditional term structure theories in that the unbiased expectations theory, the liquidity preference theory, and the market segmentation theory can all be made consistent with the model by the introduction of assumptions about the stochastic process [653].

Modern interest rate modeling is often traced to 1977 when Vasicek and Cox, Ingersoll, and Ross developed simultaneously their influential models [183, 234, 855]. Early models have fitting problems because the resulting processes may not price today's benchmark bonds correctly. An alternative approach pioneered by Ho and Lee in 1986 makes fitting the market yield curve mandatory [458]. Models based on such a paradigm are usually called **arbitrage-free** or **no-arbitrage models** [482]. The

https://doi.org/10.1017/CBO9780511546839.024 Published online by Cambridge University Press

alternatives are **equilibrium models** and Black–Scholes models, which are covered in separate chapters.

➤ **Exercise 23.1.1** A riskless security with cash flow $C_1, C_2, \ldots, C_n$ has a market price of $\sum_{i=1}^{n} C_i d(i)$. The discount factor $d(i)$ denotes the PV of \$1 at time i from now. Is the formula still valid if the cash flow depends on interest rates?

➤ **Exercise 23.1.2** Let i_t denote the period interest rate for the period from time $t-1$ to t. Assume that $1+i_t$ follows a lognormal distribution, $\ln(1+i_t) \sim N(\mu, \sigma^2)$. (1) What is the value of \$1 after n periods? (2) What are its distribution, mean, and variance?

23.2 The Binomial Interest Rate Tree

Our goal here is to construct a no-arbitrage interest rate tree consistent with the observed term structure, specifically the yields and/or yield volatilities of zero-coupon bonds of all maturities. This procedure is called **calibration**. We pick a binomial tree model in which the logarithm of the future short rate obeys the binomial distribution. The limiting distribution of the short rate at any future time is hence lognormal. In the binomial interest rate process, a binomial tree of future short rates is constructed. Every short rate is followed by two short rates for the following period. In Fig. 23.1, node A coincides with the start of period j during which the short rate r is in effect. At the conclusion of period j, a new short rate goes into effect for period $j+1$. This may take one of two possible values: r_ℓ, the "low" short-rate outcome at node B, and $r_{\rm h}$, the "high" short-rate outcome at node C. Each branch has a 50% chance of occurring in a risk-neutral economy.

We require that the paths combine as the binomial process unfolds. Suppose that the short rate r can go to $r_{\rm h}$ and r_ℓ with equal risk-neutral probability $1/2$ in a period of length Δt.[1] The volatility of $\ln r$ after Δt time is

$$\sigma = \frac{1}{2} \frac{1}{\sqrt{\Delta t}} \ln\left(\frac{r_{\rm h}}{r_\ell}\right)$$

(see Exercise 23.2.3, part (1)). Above, σ is annualized, whereas r_ℓ and $r_{\rm h}$ are period based. As

$$\frac{r_{\rm h}}{r_\ell} = e^{2\sigma\sqrt{\Delta t}}, \tag{23.1}$$

greater volatility, hence uncertainty, leads to larger $r_{\rm h}/r_\ell$ and wider ranges of possible short rates. The ratio $r_{\rm h}/r_\ell$ may change across time if the volatility is a function of time. Note that $r_{\rm h}/r_\ell$ has nothing to do with the current short rate r if σ is independent of r. The volatility of the short rate one-period forward is approximately $r\sigma$ (see Exercise 23.2.3, part (2)).

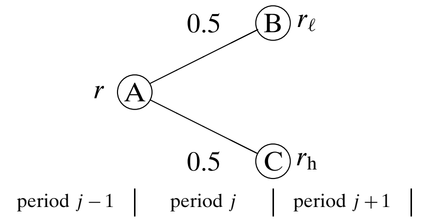

Figure 23.1: Binomial interest rate process. From node A there are two equally likely scenarios for the short rate: r_ℓ and $r_{\rm h}$. Rate r is applicable to node A in period j. Rate r_ℓ is applicable to node B and rate $r_{\rm h}$ is applicable to node C, both in period $j+1$.

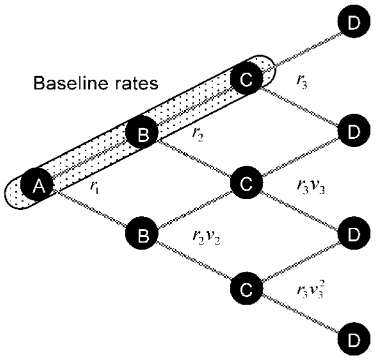

Figure 23.2: Binomial interest rate tree. The sequence at each time point shows that the short rate will converge to the lognormal distribution.

In general there are j possible rates in period j:

$$r_j,\ r_jv_j,\ r_jv_j^2,\ \dots,\ r_jv_j^{j-1},$$

where

$$v_j \equiv e^{2\sigma_j\sqrt{\Delta t}} \tag{23.2}$$

is the multiplicative ratio for the rates in period j (see Fig. 23.2). We call r_j the **baseline rates**. The subscript j in σ_j is meant to emphasize that the short rate volatility may be time dependent. In the limit, the short rate follows the following process:

$$r(t) = \mu(t)\, e^{\sigma(t)\, W(t)}, \tag{23.3}$$

in which the (percent) short-rate volatility $\sigma(t)$ is a deterministic function of time. As the expected value of $r(t)$ equals $\mu(t)\, e^{\sigma(t)^2 t/2}$, a declining short rate volatility is usually imposed to preclude the short rate from assuming implausibly high values. Incidentally, this is how the binomial interest rate tree achieves mean reversion.

One salient feature of the tree is path independence: The term structure at any node is independent of the path taken to reach it. A nice implication is that only the baseline rates r_i and the multiplicative ratios v_i need to be stored in computer memory in order to encode the whole tree. This takes up only $O(n)$ space. (Throughout this chapter, n denotes the depth of the tree, i.e., the number of discrete time periods.) The naive approach of storing the whole tree would take up $O(n^2)$ space. This can be prohibitive for large trees. For instance, modeling daily interest rate movements for 30 years amounts to keeping an array of roughly $(30 \times 365)^2/2 \approx 6 \times 10^7$ double-precision floating-point numbers. If each number takes up 8 bytes, the array would consume nearly half a gigabyte!

With the abstract process in place, the concrete numbers that set it in motion are the annualized rates of return associated with the various riskless bonds that make up the benchmark yield curve and their volatilities. In the United States, for example, the on-the-run yield curve obtained by the most recently issued Treasury securities may be used as the benchmark curve. The **term structure of (yield) volatilities** or simply the **volatility (term) structure** can be estimated from either historical data (historical volatility) or interest rate option prices such as cap prices (implied volatility) [149, 880]. The binomial tree should be consistent with both term structures. In this chapter we focus on the term structure of interest rates, deferring the handling of the volatility structure to Section 26.3.

https://doi.org/10.1017/CBO9780511546839.024 Published online by Cambridge University Press

For economy of expression, all numbers in algorithms are measured by the period instead of being annualized whenever applicable and unless otherwise stated. The relation is straightforward in the case of volatility: $\sigma(\text{period}) = \sigma(\text{annual}) \times \sqrt{\Delta t}$. As for the interest rates, consult Section 3.1.

An alternative process that also satisfies the path-independence property is the following arithmetic sequence of short rates for period j:

$$r_j,\ r_j + v_j,\ r_j + 2v_j,\ \ldots,\ r_j + (j-1)\,v_j.$$

Ho and Lee proposed this binomial interest rate model [458]. If the j possible rates for period j are postulated to be

$$ru^{j-1}, ru^{j-2}d, \ldots, rd^{j-1}$$

for some common u and d, the parameters are u and d and possibly the transition probability. This parsimonious model is due to Rendleman and Bartter [739].

➤ **Exercise 23.2.1** Verify that the variance of $\ln r$ in period k equals $\sigma_k^2(k-1)\Delta t$. (Consistent with Eq. (23.3), the variance of $\ln r(t)$ equals $\sigma(t)^2 t$ in the continuous-time limit.)

➤ **Exercise 23.2.2** Consider a short rate model such that the two equally probable short rates from the current rate r are $re^{\mu+\sigma\sqrt{\Delta t}}$ and $re^{\mu-\sigma\sqrt{\Delta t}}$, where μ may depend on time. Verify that this model can result from the binomial interest rate tree when the volatilities σ_j are all equal to some constant σ. (The μ is varied to fit the term structure.)

➤ **Exercise 23.2.3** Suppose the probability of moving from r to r_ℓ is $1-q$ and that of moving to r_{h} is q. Also assume that a period has length Δt. (1) Show that the variance of $\ln r$ after a period is $q(1-q)(\ln r_{\text{h}} - \ln r_\ell)^2$. (2) Hence, if we define σ^2 to be the above divided by Δt, then

$$\frac{r_{\text{h}}}{r_\ell} = \exp\left[\sigma\sqrt{\frac{\Delta t}{q(1-q)}}\right]$$

should replace Eq. (23.1). Now prove the variance of r after a time period of Δt is approximately $r^2\sigma^2\Delta t$.

23.2.1 Term Structure and Its Dynamics

With the binomial interest rate tree in place, the **model price** of a security can be computed by backward induction. Refer back to Fig. 23.1. Given that the values at nodes B and C are P_{B} and P_{C}, respectively, the value at node A is then

$$\frac{P_{\text{B}} + P_{\text{C}}}{2(1+r)} + \text{cash flow at node A}.$$

To save computer memory, we compute the values column by column without explicitly expanding the binomial interest rate tree (see Fig. 23.3 for illustration). Figure 23.4 contains the quadratic-time, linear-space algorithm for securities with a fixed cash flow. The same idea can be applied to any tree model.

We can compute an n-period zero-coupon bond's price by assigning \$1 to every node at period n and then applying backward induction. Repeating this step for $n = 1, 2, \ldots,$ we obtain the market discount function implied by the tree. The tree

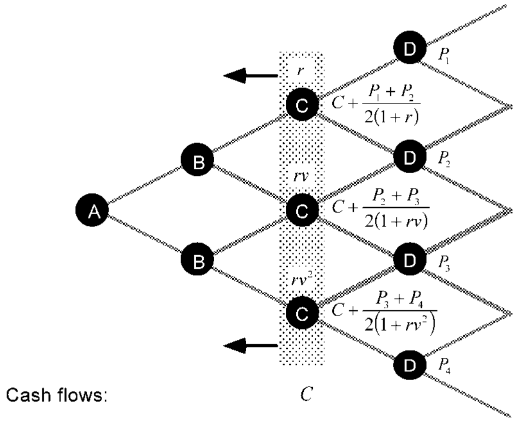

Figure 23.3: Sweep a line across time backward to compute model price.

therefore determines a term structure. Moreover, it encompasses **term structure evolution** or **dynamics** because taking any node in the tree as the current state induces a (smaller) binomial interest rate tree and, again, a term structure. The tree thus defines how the whole term structure evolves through time.

Comment 23.2.1 Suppose we want to know the m-period spot rate at time n in order to price a security whose payoff is linked to that spot rate. The tree has to be built all the way to time $n+m$ in order to obtain the said spot rate at time n, with dire performance implications. Later in Subsection 25.2.1 we will see cases in which the tree has to be built over the life of only the derivative (n periods) instead of over the life of the underlying asset ($n+m$ periods).

We shall construct interest rate trees consistent with the sample term structure in Fig. 23.5. For numerical demonstrations, we assume that the short rate volatility is such that $v \equiv r_{\text{h}}/r_{\ell} = 1.5$, independent of time.

Algorithm for model price from binomial interest rate tree:

```
input:   m, n, r[1..n], C[0..n], v[1..n];   // m ≤ n.
real     P[1..m+1];
integer i, j;
for (i = 1 to m+1) P[i] := C[m];   // Initialization.
for (i = m down to 1)   // Backward induction.
         for (j = 1 to i)
                P[j] := C[i−1] + 0.5 × (P[j] + P[j+1])/(1 + r[i] × v[i]^(j−1));
return P[1];
```

Figure 23.4: Algorithm for model price. $C[i]$ is the cash flow occurring at time i (the end of the ith period), $r[i]$ is the baseline rate for period i, $v[i]$ is the multiplicative ratio for the rates in period i, and n denotes the number of periods. Array P stores the PV at each node. All numbers are measured by the period.

https://doi.org/10.1017/CBO9780511546839.024 Published online by Cambridge University Press

Period	*1*	*2*	*3*
Spot rate (%)	4	4.2	4.3
One-period forward rate (%)	4	4.4	4.5
Discount factor	0.96154	0.92101	0.88135

Figure 23.5: Sample term structure.

An Approximate Calibration Scheme

A scheme that easily comes to mind starts with the implied one-period forward rates and then equates the expected short rate with the forward rate. This certainly works in a deterministic economy (see Exercise 5.6.6). For the first period, the forward rate is today's one-period spot rate. In general, let f_j denote the forward rate in period j. This forward rate can be derived from the market discount function by $f_j = [\, d(j)/d(j+1)\,] - 1$ (see Exercise 5.6.3). Because the ith short rate, $1 \leq i \leq j$, occurs with probability $2^{-(j-1)} \binom{j-1}{i-1}$, this means that $\sum_{i=1}^{j} 2^{-(j-1)} \binom{j-1}{i-1} r_j v_j^{i-1} = f_j$, and thus

$$r_j = \left(\frac{2}{1+v_j}\right)^{j-1} f_j. \tag{23.4}$$

The binomial interest rate tree is hence trivial to set up.

The ensuing tree for the sample term structure is shown in Fig. 23.6. For example, the price of the zero-coupon bond paying \$1 at the end of the third period is

$$\frac{1}{4} \times \frac{1}{1.04} \times \left[\frac{1}{1.0352} \times \left(\frac{1}{1.0288} + \frac{1}{1.0432}\right) + \frac{1}{1.0528} \times \left(\frac{1}{1.0432} + \frac{1}{1.0648}\right)\right] = 0.88155, \tag{23.5}$$

which is very close to, but overestimates, the discount factor 0.88135. The tree is thus not calibrated. Indeed, this bias is inherent.

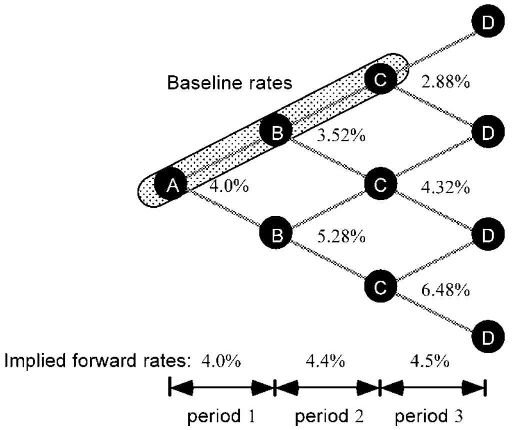

Figure 23.6: A binomial interest rate tree based on the unbiased expectations theory.

https://doi.org/10.1017/CBO9780511546839.024 Published online by Cambridge University Press

THEOREM 23.2.2 *The binomial interest rate tree constructed with Eq. (23.4) overestimates the prices of the benchmark securities in the presence of volatilities. This conclusion is independent of whether the volatility structure is matched.*

Theorem 23.2.2 implies that, under the binomial interest rate tree, the expected future spot rate exceeds the forward rate. As always, we took the money market account implicitly as numeraire. But it was argued in Subsection 13.2.1 that switching the numeraire changes the risk-neutral probability measure. Indeed, there exists a numeraire under which the forward rate equals the expected future spot rate (see Exercise 23.2.5, part (1)).

➤ **Exercise 23.2.4** (1) Prove Theorem 23.2.2 for two-period zero-coupon bonds. (2) Prove Theorem 23.2.2 in its full generality.

➤ **Exercise 23.2.5** Fix a period. (1) Show that the forward rate for that period equals the expected future spot rate under some risk-neutral probability measure. (2) Show further that the said forward rate is a martingale. (Hint: Exercise 13.2.13.)

23.2.2 Calibration of Binomial Interest Rate Trees

It is of paramount importance that the model prices generated by the binomial interest rate tree match the observed market prices. This may well be the most crucial aspect of model building. To achieve it, we can treat the backward-induction algorithm for the model price of the m-period zero-coupon bond in Fig. 23.4 as computing some function of the unknown baseline rate r_m called $f(r_m)$. A good root-finding method is then applied to solve $f(r_m) = P$ for r_m given the zero's market price P and $r_1, r_2, \ldots, r_{m-1}$. This procedure is carried out for $m = 1, 2, \ldots, n$. The overall algorithm runs in cubic time, thus hopelessly slow [508].

Calibration can be accomplished in quadratic time by the use of **forward induction** [508]. The scheme records how much \$1 at a node contributes to the model price. This number is called the **state price** as it stands for the price of a state contingent claim that pays \$1 at that particular node (state) and zero elsewhere. The column of state prices will be established by moving *forward* from time 1 to time n.

Let us be more precise. Suppose we are at time j and there are $j+1$ nodes. Let the baseline rate for period j be $r \equiv r_j$, the multiplicative ratio be $v \equiv v_j$, and $P_1, P_2, \ldots, P_j$ be the state prices a period prior, corresponding to rates $r, rv, \ldots, rv^{j-1}$. By definition, $\sum_{i=1}^{j} P_i$ is the price of the $(j-1)$-period zero-coupon bond. One dollar at time j has a known market value of $1/[\,1 + S(j)\,]^j$, where $S(j)$ is the j-period spot rate. Alternatively, this dollar has a PV of

$$g(r) \equiv \frac{P_1}{(1+r)} + \frac{P_2}{(1+rv)} + \frac{P_3}{(1+rv^2)} + \cdots + \frac{P_j}{(1+rv^{j-1})}.$$

So we solve

$$g(r) = \frac{1}{[\,1 + S(j)\,]^j} \tag{23.6}$$

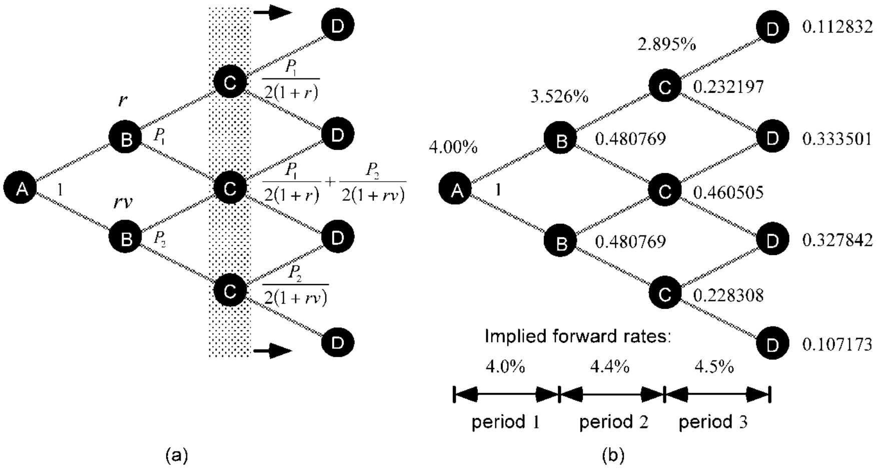

Figure 23.7: Sweep a line forward to compute binomial state price tree. (a) The state price at a node is a weighted sum of the state prices of its two predecessors. (b) The binomial state price tree calculated from the sample term structure and the resulting calibrated binomial interest rate tree. It prices the benchmark bonds correctly.

for r. Given a decreasing market discount function, a unique positive solution for r is guaranteed.[2] The state prices at time j can now be calculated as

$$\frac{P_1}{2(1+r)}, \frac{P_1}{2(1+r)}+\frac{P_2}{2(1+rv)}, \ldots, \frac{P_{j-1}}{2\,(1+rv^{j-2})}$$
$$+\frac{P_j}{2\,(1+rv^{j-1})}, \frac{P_j}{2\,(1+rv^{j-1})}$$

(see Fig. 23.7(a)). We call a tree with these state prices a **binomial state price tree**. Figure 23.7(b) shows one such tree. The calibrated tree is shown in Fig. 23.8.

The Newton–Raphson method can be used to solve for the r in Eq. (23.6) as $g'(r)$ is easy to evaluate. The monotonicity and the convexity of $g(r)$ also facilitate root finding. A good initial approximation to the root may be provided by Eq. (23.4), which

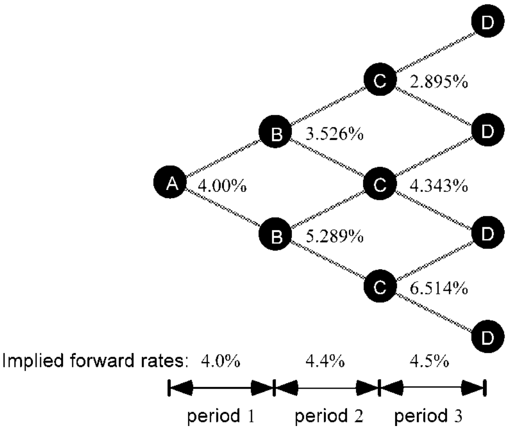

Figure 23.8: Calibrated binomial interest rate tree. This tree is from Fig. 23.7(b).

https://doi.org/10.1017/CBO9780511546839.024 Published online by Cambridge University Press

Algorithm for building calibrated binomial interest rate tree:

input: n, $S[\,1..n\,]$, $\sigma[\,1..n\,]$;
real $P[\,0..n\,]$, $r[\,1..n\,]$, r, v;
integer i, j;
$P[\,0\,] := 0$; // Dummy variable; remains zero throughout.
$P[\,1\,] := 1$;
$r[\,1\,] := S[\,1\,]$;
for ($i = 2$ to n) {
  $v := \exp[\,2 \times \sigma[\,i\,]\,]$;
  $P[\,i\,] := 0$;
  for ($j = i$ down to 1) // State prices at time $i-1$.
    $P[\,j\,] := \frac{P[\,j-1\,]}{2\times(1+r[\,i-1\,]\,v^{j-2})} + \frac{P[\,j\,]}{2\times(1+r[\,i-1\,]\,v^{j-1})}$;
  Solve $\sum_{j=1}^{i} \frac{P[\,j\,]}{(1+rv^{j-1})} = (1+S[\,i\,])^{-i}$ for r;
  $r[\,i\,] := r$;
}
return $r[\,]$;

Figure 23.9: Algorithm for building calibrated binomial interest rate tree. $S[\,i\,]$ is the i-period spot rate, $\sigma[\,i\,]$ is the percent volatility of the rates for period i, and n is the number of periods. All numbers are measured by the period. The period-i baseline rate is computed and stored in $r[\,i\,]$.

is guaranteed to underestimate the root (see Theorem 23.2.2). Using the previous baseline rate as the initial approximation to the current baseline rate also works well.

The preceding idea is straightforward to implement (see Fig. 23.9). The total running time is $O(\mathcal{C}n^2)$, where $\mathcal{C}$ is the maximum number of times the root-finding routine iterates, each consuming $O(n)$ work. With a good initial guess, the Newton–Raphson method converges in only a few steps [190, 625].

Let us follow up with some numerical calculations. One dollar at the end of the second period should have a PV of 0.92101 according to the sample term structure. The baseline rate for the second period, r_2, satisfies

$$\frac{0.480769}{1+r_2} + \frac{0.480769}{1+1.5\times r_2} = 0.92101.$$

The result is $r_2 = 3.526\%$. This is used to derive the next column of state prices shown in Fig. 23.7(b) as 0.232197, 0.460505, and 0.228308, whose sum gives the correct market discount factor 0.92101. The baseline rate for the third period, r_3, satisfies

$$\frac{0.232197}{1+r_3} + \frac{0.460505}{1+1.5\times r_3} + \frac{0.228308}{1+(1.5)^2\times r_3} = 0.88135.$$

The result is $r_3 = 2.895\%$. Now, redo Eq. (23.5) using the new rates:

$$\frac{1}{4}\times\frac{1}{1.04}\times\left[\frac{1}{1.03526}\times\left(\frac{1}{1.02895}+\frac{1}{1.04343}\right)+\frac{1}{1.05289}\right.$$
$$\left.\times\left(\frac{1}{1.04343}+\frac{1}{1.06514}\right)\right] = 0.88135,$$

$$
\begin{tikzcd}
 & & {[\,0.971865\,]} \\
 & {\begin{bmatrix} 0.965941 \\ 0.932250 \end{bmatrix}} \arrow[ur] \arrow[dr] & \\
{\begin{bmatrix} 0.96154 \\ 0.92101 \\ 0.88135 \end{bmatrix}} \arrow[ur] \arrow[dr] & & {[\,0.958378\,]} \\
 & {\begin{bmatrix} 0.949767 \\ 0.900959 \end{bmatrix}} \arrow[ur] \arrow[dr] & \\
 & & {[\,0.938844\,]}
\end{tikzcd}
$$

Figure 23.10: Term structure dynamics. Each node lists the market discount function in increasing maturities.

an exact match. The tree in Fig. 23.8 therefore prices without bias the benchmark securities. The term structure dynamics of the calibrated tree is shown in Fig. 23.10.

➤ **Exercise 23.2.6** (1) Based on the sample term structure and its associated binomial interest rate tree in Fig. 23.8, what is the next baseline rate if the four-period spot rate is 4.4%? (2) Confirm Theorem 23.2.2 by demonstrating that the baseline rate produced by Eq. (23.4) is smaller than the one derived in (1).

➤ **Exercise 23.2.7** (1) Suppose we are given a binomial state price tree and wish to price a security with the payoff function c at time j by using the risk-neutral pricing formula $d(j)\,E[\,c\,]$. What is the probability of each state's occurring at time j? (2) Take the binomial state price tree in Fig. 23.7(b). What are the probabilities of the C nodes in this risk-neutral economy?

➤ **Exercise 23.2.8** Compute the n discount factors implied by the tree in $O(n^2)$ time.

➤ **Exercise 23.2.9** Start with a binomial interest rate tree but *without* the branching probabilities, such as Fig. 23.2. (1) Suppose the state price tree is also given. (2) Suppose only the state prices at the terminal nodes are given and assume that the path probabilities for all paths reaching the same node are equal. How do we calculate the branching probabilities at each node in either case? (The result was called the implied binomial tree in Exercise 9.4.3.)

➢ **Programming Assignment 23.2.10** Program the algorithm in Fig. 23.9 with the Newton–Raphson method.

➢ **Programming Assignment 23.2.11** Calibrate the tree with the secant method.

23.3 Applications in Pricing and Hedging

23.3.1 Spread of Nonbenchmark Option-Free Bonds

Model prices calculated by the calibrated tree as a rule do not match market prices of nonbenchmark bonds. To gauge the incremental return, or **spread**, over the benchmark bonds, we look for the spread that, when added uniformly over the short rates in the tree, makes the model price equal the market price. Obviously the spread of a benchmark security is zero. We apply the spread concept to option-free bonds first

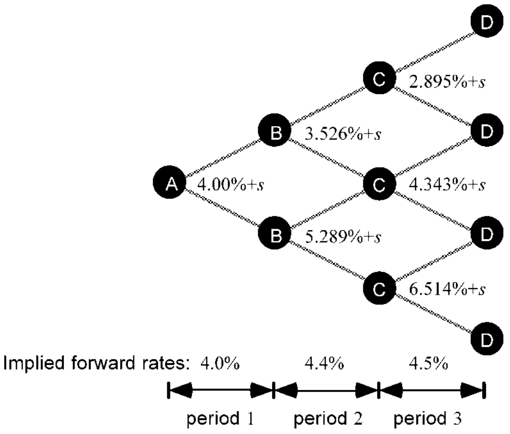

Figure 23.11: Spread over short rates of binomial interest rate tree. This tree is constructed from the calibrated binomial interest rate tree in Fig. 23.8 by the addition of a constant spread s to each short rate in the tree.

and return to bonds that incorporate embedded options in Subsection 27.4.3. The techniques are identical save for the possibility of early exercise.

It is best to illustrate the idea with an example. Start with the tree in Fig. 23.11. Consider a security with cash flow C_i at time i for $i = 1, 2, 3$. Its model price is

$$\begin{aligned} p(s) \equiv \frac{1}{1.04+s} \times \Bigg\{ C_1 + \frac{1}{2} \times \frac{1}{1.03526+s} \\ \times \left[C_2 + \frac{1}{2} \left(\frac{C_3}{1.02895+s} + \frac{C_3}{1.04343+s} \right) \right] \\ + \frac{1}{2} \times \frac{1}{1.05289+s} \times \left[C_2 + \frac{1}{2} \left(\frac{C_3}{1.04343+s} + \frac{C_3}{1.06514+s} \right) \right] \Bigg\}. \end{aligned}$$

Given a market price of P, the spread is the s that solves $P = p(s)$.

In general, if we add a constant amount s to every rate in the binomial interest rate tree, the model price will be a monotonically decreasing, convex function of s. Call this function $p(s)$. For a market price P, we use the Newton–Raphson root-finding method to solve $p(s) - P = 0$ for s. However, a quick look at the preceding equation reveals that evaluating $p'(s)$ directly is infeasible. Fortunately the tree can be used to evaluate both $p(s)$ and $p'(s)$ during backward induction. Here is the idea. Consider an arbitrary node A in the tree associated with the short rate r. In the process of computing the model price $p(s)$, a price $p_{\text{A}}(s)$ is computed at A. Prices computed at A's two successor nodes, B and C, are discounted by $r + s$ to obtain $p_{\text{A}}(s)$:

$$p_{\text{A}}(s) = c + \frac{p_{\text{B}}(s) + p_{\text{C}}(s)}{2(1 + r + s)},$$

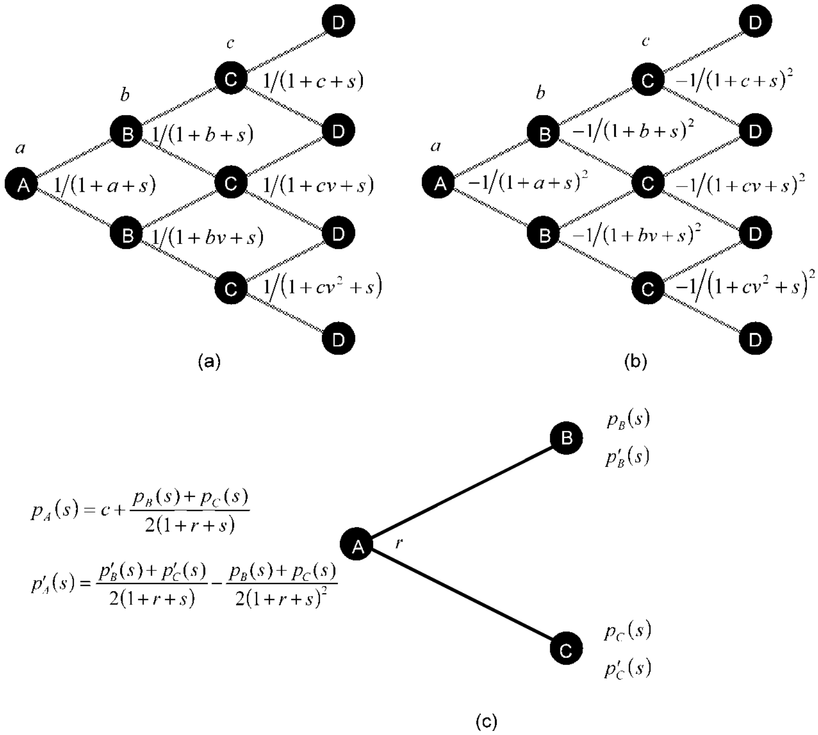

Figure 23.12: The differential tree method. (a) The original binomial interest rate tree with the short rates replaced with the discount factors, (b) the derivatives of the numbers on the tree, and (c) the simultaneous evaluation of a function and its derivative on the tree by use of the numbers from (a) and (b).

where c denotes the cash flow at A. To compute $p'_{\mathrm{A}}(s)$ as well, node A calculates

$$p'_{\mathrm{A}}(s) = \frac{p'_{\mathrm{B}}(s) + p'_{\mathrm{C}}(s)}{2(1+r+s)} - \frac{p_{\mathrm{B}}(s) + p_{\mathrm{C}}(s)}{2(1+r+s)^2}, \tag{23.7}$$

which is easy if $p'_{\mathrm{B}}(s)$ and $p'_{\mathrm{C}}(s)$ are also computed at nodes B and C. Applying the preceding procedure inductively will eventually lead to $p(s)$ and $p'(s)$ at the root. See Fig. 23.12 for illustration. This technique, which is due to Lyuu, is called the **differential tree method** and has wide applications [625]. It is also related to **automatic differentiation** in numerical analysis [602, 687].

Let us analyze the differential tree algorithm in Fig. 23.13. Given a spread, step 1 computes the PV, step 2 computes the derivative of the PV according to Eq. (23.7), and step 3 implements the Newton–Raphson method for the next approximation. If $\mathcal{C}$ represents the number of times the tree is traversed, which takes $O(n^2)$ time, the total running time is $O(\mathcal{C}n^2)$. In practice, $\mathcal{C}$ is a small constant. The memory requirement is $O(n)$.

Now we go through a numerical example. Consider a 3-year 5% bond with a market price of 100.569. For simplicity, assume that the bond pays annual interest. The spread can be shown to be 50 basis points over the tree (see Fig. 23.14). For comparison, let us compute the yield spread and the static spread of the nonbenchmark bond over an otherwise identical benchmark bond. Recall that the static spread is

Algorithm for computing spread based on differential tree method:

```
input:    n, P, r[1..n], C[0..n], v[1..n], ϵ;
real      P[1..n+1], P'[1..n+1], s_old, s_new;
integer   i, j;
s_new := 0;  // Initial guess.
P[1] := ∞;
while [ | P[1] − P | > ϵ ] {
        for (i = 1 to n+1) { P[i] := C[n]; P'[i] := 0; }
        s_old := s_new;
        for (i = n down to 1)   // Sweep the column backward in time.
            for (j = 1 to i) {
                1. P[j] := C[i−1] + (P[j] + P[j+1])/(2 × (1 + r[i] × v[i]^(j−1) + s_old));
                2. P'[j] := (P'[j] + P'[j+1])/(2 × (1 + r[i] × v[i]^(j−1) + s_old)) −
                     (P[j] + P[j+1])/(2 × (1 + r[i] × v[i]^(j−1) + s_old))^2;
            }
        3. s_new := s_old − (P[1] − P)/P'[1];   // Newton-Raphson.
}
return s_new;
```

Figure 23.13: Algorithm for computing spread based on differential tree method. P is the market price, $r[\,i\,]$ is the baseline rate for period i, $C[\,i\,]$ contains the cash flow at time i, $v[\,i\,]$ is the multiplicative ratio for the rates in period j, and n is the number of periods. All numbers are measured by the period. The prices and their derivatives are stored in $P[\,]$ and $P'[\,]$, respectively.

the incremental return over the spot rate curve, whereas the spread based on the binomial interest rate tree is one over the future short rates. The yield to maturity of the nonbenchmark bond can be calculated to be 4.792%. The 3-year Treasury has a market price of

$$\frac{5}{1.04}+\frac{5}{(1.042)^2}+\frac{105}{(1.043)^3}=101.954 \tag{23.8}$$

and a yield to maturity of 4.292%. The yield spread is thus $4.792\% - 4.292\% = 0.5\%$. The static spread can also be found to be 0.5%. So all three spreads turn out to be 0.5% up to round-off errors.

➤ **Exercise 23.3.1** Does the idea of spread assume parallel shifts in the term structure?

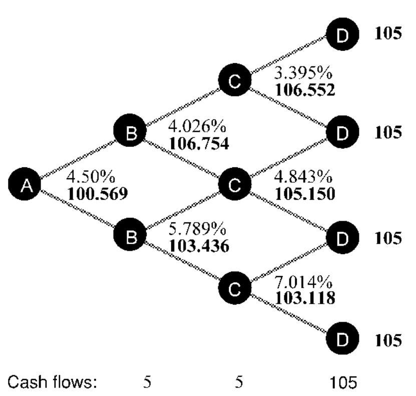

Figure 23.14: Price tree with spread. Based on the tree in Fig. 23.11, the price tree is computed for a 3-year bond paying 5% annual interest. Each node of the tree signifies, besides the short rate, the discounted value of its future cash flows plus the cash flow at that node, if any. The model price is 100.569.

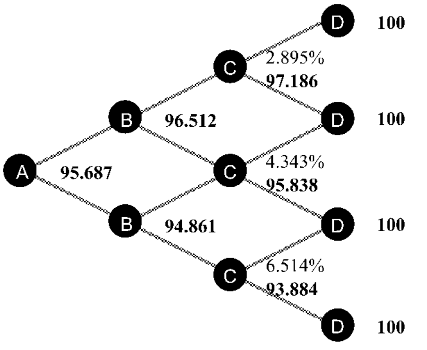

Figure 23.15: Futures price. The price tree is computed for a 2-year futures contract on a 1-year T-bill. C nodes store, besides the short rate, the discounted values of the 1-year T-bill under the model. A and B nodes calculate the expected values. The futures price is 95.687.

➢ **Programming Assignment 23.3.2** Implement the algorithm in Fig. 23.13.

➢ **Programming Assignment 23.3.3** Implement a differential tree method for the implied volatility of American options under the BOPM [172, 625].

23.3.2 Futures Price

The futures price is a martingale under the risk-neutral probability (see Exercise 13.2.11). To compute it, we first use the tree to calculate the underlying security's prices at the futures contract's delivery date to which the futures price converges. Then we find the expected value. In Fig. 23.15, for example, we are concerned with a 2-year futures contract on a 1 year T-bill. The futures price is found to be 95.687. The futures price can be computed in $O(n)$ time.

If the contract specification for a futures contract does not call for a quote that equals the result of our computation, steps have to be taken to convert the theoretical value into one consistent with the specification. The theoretical value above, for instance, corresponds to the invoice price of the T-bill futures traded on the CBT, but it is the index price that gets quoted.

➤ **Exercise 23.3.4** (1) How do we compute the forward price for a forward contract on a bond? (2) Calculate the forward price for a 2-year forward contract on a 1-year T-bill.

23.3.3 Fixed-Income Options

Determining the values of fixed-income options with a binomial interest rate tree follows the same logic as that of the binomial tree algorithm for stock options in Chap. 9. Hence only numerical examples are attempted here. Consider a 2-year 99 European call on the 3-year 5% Treasury. Assume that the Treasury pays annual interest. From Fig. 23.16 the 3-year Treasury's price minus the $5 interest could be $102.046, $100.630, or $98.579 2 years from now. Because these prices do not include the accrued interest, we should compare the strike price against them. The call is therefore in the money in the first two scenarios, with values of $3.046 and $1.630, and out of the money in the third scenario. The option value is calculated to be $1.458 in Fig. 23.16(a). European interest rate puts can be valued similarly. Consider a 2-year 99 European put on the same security. At expiration, the put is in the money only if the Treasury is worth $98.579 without the accrued interest. The option value is computed to be $0.096 in Fig. 23.16(b).

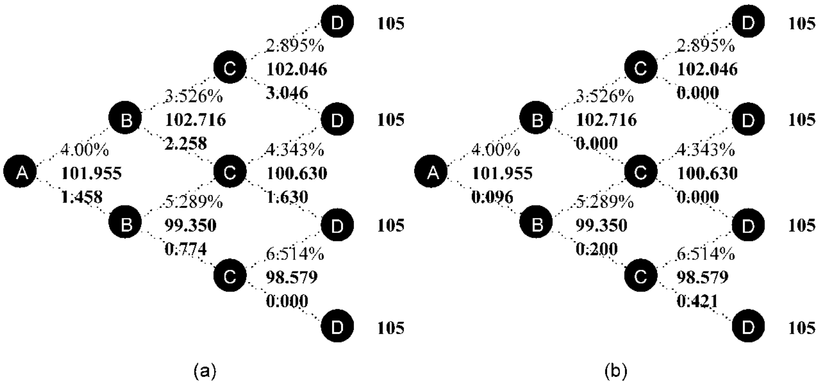

Figure 23.16: European options on Treasuries. The above price trees are computed for the 2-year 99 European (a) call and (b) put on the 3-year 5% Treasury. Each node of the tree signifies the short rate, the Treasury price without the \$5 interest (except the D nodes), and the option value. The price \$101.955 is slightly off compared with Eq. (23.8) because of round-off errors.

If the option is American and the underlying bond generates payments before the option's expiration date, early exercise needs to be considered. The criterion is to compare the intrinsic value against the option value at each node. The details are left to the reader.

The PV of the strike price is $\text{PV}(X) = 99 \times 0.92101 = 91.18$. The Treasury is worth $B = 101.955$. The PV of the interest payments during the life of the options is

$$\text{PV}(I) = 5 \times 0.96154 + 5 \times 0.92101 = 9.41275.$$

The call and the put are worth $C = 1.458$ and $P = 0.096$, respectively. Hence

$$C = P + B - \text{PV}(I) - \text{PV}(X).$$

The put–call parity is preserved.

➤ **Exercise 23.3.5** Prove that an American option on a zero-coupon bond will not be exercised early.

➤ **Exercise 23.3.6** Derive the put–call parity for options on coupon bonds.

➢ **Programming Assignment 23.3.7** Write a program to price European options on the Treasuries.

23.3.4 Delta (Hedge Ratio)

It is important to know how much the option price changes in response to changes in the price of the underlying bond. This relation is called the delta (or hedge ratio), defined as

$$\frac{O_{\text{h}} - O_{\ell}}{P_{\text{h}} - P_{\ell}}.$$

P_{h} and P_{ℓ} denote the bond prices if the short rate moves up and down, respectively. Similarly, O_{h} and O_{ℓ} denote the option values if the short rate moves up and down, respectively. Because delta measures the sensitivity of the option value to changes in

Short rate	Horizon price	Probability
2.895%	0.971865	0.25
4.343%	0.958378	0.50
6.514%	0.938844	0.25

Figure 23.17: HPRs. The horizon is two periods from now.

the underlying bond price, it shows how to hedge one with the other [84]. Take the call and put in Fig. 23.16 as examples. Their deltas are

$$\frac{0.774 - 2.258}{99.350 - 102.716} = 0.441, \qquad \frac{0.200 - 0.000}{99.350 - 102.716} = -0.059,$$

respectively.

23.3.5 Holding Period Returns

Analyzing the holding period return (HPR) with the binomial interest rate tree is straightforward. As an example, consider a two-period horizon for three-period zero-coupon bonds. Based on the price dynamics in Fig. 23.10, the HPRs are obtained in Fig. 23.17. If the bonds are coupon bearing, the interim cash flows should be reinvested at the prevailing short rate and added to the horizon price. The probability distribution of the scenario analysis is provided by the model, not exogenously.

23.4 Volatility Term Structures

The binomial interest rate tree can be used to calculate the yield volatility of zero-coupon bonds. Consider an n-period zero-coupon bond. First find its yield to maturity y_{h} (y_ℓ, respectively) at the end of the initial period if the rate rises (declines, respectively). The yield volatility for our model is defined as $(1/2)\ln(y_{\mathrm{h}}/y_\ell)$. For example, based on the tree in Fig. 23.8, the 2-year zero's yield at the end of the first period is 5.289% if the rate rises and 3.526% if the rate declines. Its yield volatility is therefore

$$\frac{1}{2}\ln\left(\frac{0.05289}{0.03526}\right) = 20.273\%.$$

Now consider the 3-year zero-coupon bond. If the rate rises, the price of the zero 1 year from now will be

$$\frac{1}{2} \times \frac{1}{1.05289} \times \left(\frac{1}{1.04343} + \frac{1}{1.06514}\right) = 0.90096.$$

Thus its yield is

$$\sqrt{\frac{1}{0.90096}} - 1 = 0.053531.$$

If the rate declines, the price of the zero 1 year from now will be

$$\frac{1}{2} \times \frac{1}{1.03526} \times \left(\frac{1}{1.02895} + \frac{1}{1.04343}\right) = 0.93225.$$

Thus its yield is

$$\sqrt{\frac{1}{0.93225}}-1=0.0357.$$

The yield volatility is hence

$$\frac{1}{2}\ln\left(\frac{0.053531}{0.0357}\right)=20.256\%,$$

slightly less than the 1-year yield volatility. Interestingly, this is consistent with the reality that longer-term bonds typically have lower yield volatilities than shorter-term bonds. The procedure can be repeated for longer-term zeros to obtain their yield volatilities.

We started with v_i and then derived the volatility term structure. In practice, the steps are reversed. The volatility term structure is supplied by the user along with the term structure. The v_is – hence the short rate volatilities by Eq. (23.2) – and the r_is are then simultaneously determined. The result is the Black–Derman–Toy model, which is covered in Section 26.3.

Suppose the user supplies the volatility term structure that results in $(v_1, v_2, v_3, \dots)$ for the tree. The volatility term structure one period from now will be determined by $(v_2, v_3, v_4, \dots)$, not $(v_1, v_2, v_3, \dots)$. The volatility term structure supplied by the user is hence not maintained through time.

➤ **Exercise 23.4.1** Suppose we add a binomial process for the stock price to our binomial interest rate model. In other words, stock price S can in one period move to Su or Sd. What are the constraints on u and d?

➢ **Programming Assignment 23.4.2** Add the annualized term structure of yield volatilities to the output of the program of Programming Assignment 23.2.10.

NOTES

1. By designating the risk-neutral probabilities as $1/2$, we are obliged to adjust the state variable, the short rate, in order to match the desired distribution. This was done in Exercise 9.3.1, for example, in the case of the BOPM. An alternative is to prescribe the state variable's values on the tree and then to find the probabilities. This was the approach of the finite-difference method in Section 18.1.
2. This is because $g(r)$ is strictly decreasing with $g(0)=\sum_{i=1}^{j} P_i > 1/[\,1+S(j)\,]^j$ and $g(\infty)=0$.