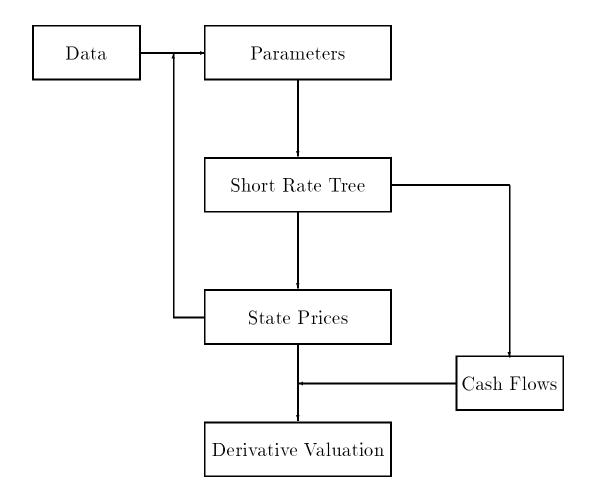
Backus & Zin/April 1, 1999

#### Binomial Models 1

- 1. Flow chart
- 2. Rate trees
- 3. Contingent claims and state prices
- 4. Valuation 1: one period at a time
- 5. Valuation 2: all at once
- 6. Models: Ho and Lee, Black-Derman-Toy
- 7. Calibration of parameters
- 8. Examples of asset valuation
- 9. Eurodollar options (term structure of volatility revisited)
- 10. Summary and final thoughts

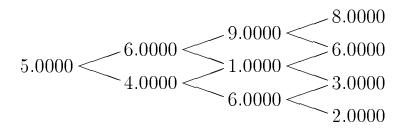
#### 1. Flow Chart



• We'll start in the middle (how a rate tree turns into state prices) and work out from there

#### 2. A Short Rate Tree

• Consider this tree for one-period interest rates ("short rates"):



- Key feature: (down,up) and (up,down) get you to the same place (the math term is "lattice")
- Convention (arbitrary, but you have to choose something):
  - With continuous compounding, the one-period discount factor satisfies

$$b^1 = \exp(-rh/100)$$
  
 $r = -(100/h)\log(b^1)$ 

- If data differ, convert to this basis (note that  $b^1$  is convention free)
- (One-period) discount factor tree (h = 1/4):

$$0.9876 < \underbrace{\begin{array}{c} 0.9851 \\ 0.9975 \\ 0.9900 \\ \end{array}}_{0.9851} \underbrace{\begin{array}{c} 0.9802 \\ 0.9951 \\ 0.9925 \\ 0.9950 \\ \end{array}}_{0.9950}$$

(ie, this is the tree for  $b^1$ )

#### 3. States

- Terminology:
  - "state" means a scenario or situation
  - state-contingent claims, or derivatives, are assets whose cash flows depend on the situation at a future date (eg, option payoffs depend on the future value of the underlying)
- In binomial models, the state is the location in the tree
- Label the location in the tree by (i, n):

i = number of up moves since start

n = number of periods since start

# Examples:

- -(i,n) = (0,0) is the initial node
- -(i,n)=(3,3) is the upper right node on the preceding page
- Note the extra dimension:
  - discount factors value cash flows at different dates
  - here we distinguish by "state" as well as date (ie, we introduce uncertainty)
- For valuation we need
  - a list of states, the ordered pair (i, n)
  - the cash flows associated with each state, c(i, n)
  - state prices: the value of one dollar in each state, Q(i,n)

# 4. Recursive Valuation: Theory

• Apply these equations at each node (i, n), starting at the end:

$$q_{u} = \pi_{u}^{*}b^{1}$$

$$q_{d} = \pi_{d}^{*}b^{1} = (1 - \pi_{u}^{*})b^{1}$$

$$p = c + q_{u}p_{u} + q_{d}p_{d}$$

where

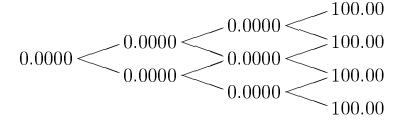
- (i+1,n+1) is the "up" state and (i,n+1) the "down"
- $-q_u(q_d)$  is the value of one dollar in the up (down) state
- $-p_u(p_d)$  is the price of the asset in the up (down) state
- $-\pi_{u}^{*}(\pi_{d}^{*})$  is the risk-neutral probability of the up (down) state
- -c is the asset's cash flow in (i, n)
- -p is the asset's price in (i, n)

# • State prices

- discount future cash flows: that's the role of  $b^1$
- adjust for risk: the risk-neutral probabilities might be called "risk-adjusted" probabilities
- divide the discount factor:  $q_u + q_d = b^1$
- The 50-50 rule: set  $\pi_u^* = \pi_d^* = 0.5$  (completely arbitrary, but absolutely standard)

# 5. Recursive Valuation: Examples

- Example 1: 3-period zero
  - Cash flows are



- Prices at the end are

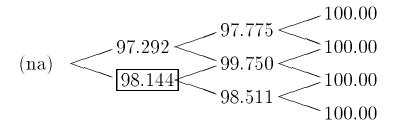
- Find prices one period from the end:

- Details for "boxed" node (0,2):
  - \* state prices are  $q_u = q_d = 0.9851/2 = 0.4926$
  - \* zero's price is

$$p\ =\ 0+0.4926(100+100)\ =\ 98.511$$

# 5. Recursive Valuation: Examples (continued)

- Example 1: 3-period zero (continued)
  - Find prices two periods from the end:



- Details for "boxed" node (0,1):
  - \* state prices are  $q_u = q_d = 0.9900/2 = 0.4950$
  - \* zero's price is

$$p = 0 + 0.4950(99.750 + 98.511) = 98.144$$

- Find price for initial node:

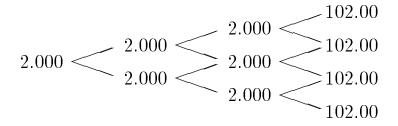
$$\begin{array}{c|c}
97.292 & 97.775 & 100.00 \\
98.144 & 99.750 & 100.00 \\
98.511 & 100.00
\end{array}$$

- Details for "boxed" node (0,0):
  - \* state prices are  $q_u = q_d = 0.9876/2 = 0.4938$
  - \* zero's price is

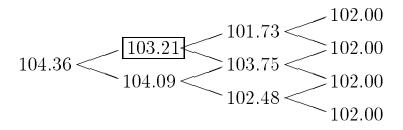
$$p = 0 + 0.4938(97.292 + 98.144) = 96.504$$

### 5. Recursive Valuation: Examples (continued)

- Example 2: 3-period 8% bond (quarterly payments)
  - Cash flows are



- Price path for bond:



- Details for "boxed" node (1,1):
  - \* state prices are  $q_u = q_d = 0.9851/2 = 0.4926$
  - \* zero's price is

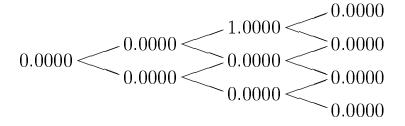
$$p = 2 + 0.4926(101.73 + 103.75) = 103.21$$

(note the cash flow of 2 here)

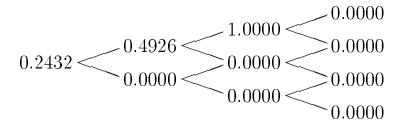
- Same approach: we can value anything!

# 5. Recursive Valuation: Examples (continued)

- Example 3: one dollar in state (2,2) (pure state-contingent claim)
  - Cash flows are



- Price path is:



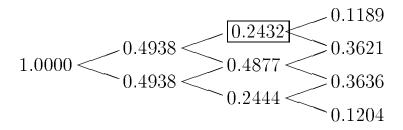
- Details for initial node (0,0):
  - \* state prices are  $q_u = q_d = 0.9876/2 = 0.4938$
  - \* zero's price is

$$p = 0 + 0.4938(0.4926 + 0) = 0.2432$$

- Comment: this example is a little abstract, but it turns out to be useful

#### 6. All-at-Once Valuation

- A second approach: multiply state prices by cash flows and add
- State prices Q(i, n) for our environment are:



#### Comments:

- these are prices now for one dollar payable in the relevant state/node (think about this: it's not a price path)
- initial node: a dollar now is worth a dollar
- the node with the box is example 3
- we'll see shortly where these come from
- Example 1:

$$p = 100 \times (0.1189 + 0.3621 + 0.3636 + 0.1204) = 96.50$$
 (same answer by different route)

• Discount factors and spot rates:

$$b^{n} = \sum_{i} Q(i, n)$$
  

$$b^{n+1} = \sum_{i} Q(i, n)b^{1}(i, n)$$
  

$$y^{n} = -(100/nh) \log b^{n}$$

### 7. Computing State Prices

• Duffie's formula:

$$q(i,n+1) = \begin{cases} \pi_d^*b^1(i,n)Q(i,n) & \text{if } i = 0 \\ \pi_d^*b^1(i,n)Q(i,n) + \\ \pi_u^*b^1(i-1,n)Q(i-1,n) & \text{if } 0 < i < n+1 \\ \pi_u^*b^1(i-1,n)Q(i-1,n) & \text{if } i = n+1 \end{cases}$$

#### • Comments:

- the idea is to compute all the state prices at once, starting at the beginning
- saves a lot of work
- on the edges (the first and third lines): price is  $q_u$  ( $q_d$ ) times the current state price
- in the middle (second line): since you can reach the node from two previous nodes, it has two components
- try a few steps to see how it works
- Chriss (*Black-Scholes and Beyond*, ch 6) has a nice summary (he calls them Arrow-Debreu prices)

# • Discount factors and spot rates:

#### 8. Models

- A model is a rule for generating a short rate tree (once we have the tree, we know how to do the rest)
- The Ho and Lee model:
  - Short rate rule:

$$r_{t+1} = r_t + \mu_{t+1} + h^{1/2} \sigma \varepsilon_{t+1},$$

with

$$\varepsilon_{t+1} = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

(h converts  $\sigma$  to annual units)

- Properties:
  - \* The mean change in r is

$$E_t(r_{t+1} - r_t) = \mu_{t+1} + h^{1/2}\sigma \left[ (1/2)(1) + (1/2)(-1) \right]$$
  
=  $\mu_{t+1}$ 

\* The variance of the change in r is

$$Var_t(r_{t+1} - r_t) = h\sigma^2 \left[ (1/2)(1)^2 + (1/2)(-1)^2 \right]$$
  
=  $h\sigma^2$ 

\* A discrete approximation to Vasicek without mean reversion  $(\varphi = 1)$ 

# 8. Models (continued)

- The logarithmic model (Ho and Lee in logs)
  (Tuckman calls this the "original Salomon model")
  - Let  $z = \log r$  [so that  $r = \exp(z)$ ]:  $z_{t+1} = z_t + \mu_{t+1} + h^{1/2} \sigma \varepsilon_{t+1}$

with

$$\varepsilon_{t+1} = \begin{cases}
+1 & \text{with probability } 1/2 \\
-1 & \text{with probability } 1/2
\end{cases}$$

- Comments:
  - $* \log r$  keeps r positive
  - \* the volatility parameter  $\sigma$  applies to the rate, hence consistent with industry practice for quoting implied volatilities

# 8. Models (continued)

- The Black-Derman-Toy model (logarithmic model with time-dated volatility)
  - Black-Derman-Toy model  $(z = \log r)$ :

$$z(i,n) = z(0,0) + \sum_{j=1}^{n} \mu_{t+j} + (2i-n)h^{1/2}\sigma_n$$

Find short rates from  $r(i, n) = \exp[z(i, n)]$ 

- What?
  - \* Ho and Lee might be expressed as

$$r(i,n) = r(0,0) + \sum_{j=1}^{n} \mu_{t+j} + (2i-n)h^{1/2}\sigma$$

\* The last term: in state (i, n), we have experienced i up moves over n periods. Apparently we experienced n - i down moves, too, so the total effect of up and down moves is

$$i - (n - i) = 2i - n$$

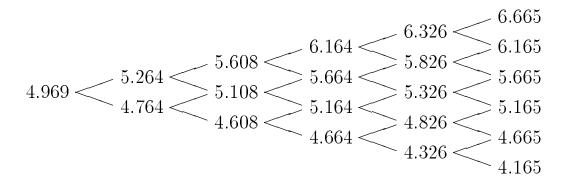
- \* BDT: change to logs and make  $\sigma$  depend on time
- \* Why is this clever? If we did this through the usual route, (up,down) and (down,up) wouldn't end up in the same place if  $\sigma$  isn't the same each period. BDT finesse this by defining up and down relative to the mean rate in that period (ie, by the horizontal difference between rates in the tree).
- \* Why is this useful? Because the term structure of volatility isn't flat.

### 9. Choosing Parameters

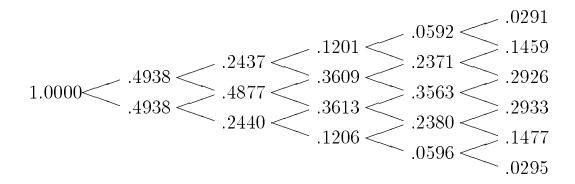
- Choosing volatilities  $\sigma$ :
  - estimate from recent data (eg, standard deviation of changes in spot rates)
  - infer from option prices (interest rate caps, eurodollar futures, swaptions)
- Choosing "drift" parameters  $\mu$ :
  - reproduce current spot rates exactly!
  - remark: absolutely essential (how can you value options if the spot rates are wrong?)
  - Duffie's formula is extremely helpful: quick way to compute spot rates for a model, so they can be compared to the data
  - algorithm:
    - 1. guess  $\mu$ 's
    - 2. compute rate tree
    - 3. use Duffie's formula to compute spot rates
    - 4. compare spot rates to data
    - 5. Choose:
      - \* if spot rates in the model are the same as the data, you're done
    - \* if they're different, return to 1 with a new guess (this is clearer if you run through it on a spreadsheet)

### 9. Choosing Parameters (continued)

- Calibrating the Ho and Lee model
  - Set h = 0.25 (3-month eurodollar contracts coming up)
  - Choose  $\sigma = 0.5\%$  (ballpark number, more later)
  - Current spot rates are (4.969, 4.991, 5.030, 5.126, 5.166, 5.207) (these match eurodollar futures prices, an issue we can explore in more depth later if you like)
- The resulting interest rate tree is



• State prices (courtesy of Duffie's formula):



# 9. Choosing Parameters (continued)

- Calibrating the Ho and Lee model (continued)
  - Verifying spot rates:

$$b^{1} = 0.4938 + 0.4938 = 0.9877$$

$$\Rightarrow y^{1} = -(100/h) \log b^{1} = 4.969$$

$$b^{4} = 0.0592 + 0.2371 + 0.3563 + 0.2380 + 0.0596 = 0.9375$$

$$\Rightarrow y^{4} = -(100/4h) \log b^{4} = 5.126$$

(you need more digits to reproduce this exactly)

- Complete table of discount factors and spot rates

Maturity	Discount Factor	Spot Rate
0.25	0.9877	4.969
0.50	0.9754	4.991
0.75	0.9630	5.030
1.00	0.9500	5.126
1.25	0.9375	5.166
1.50	0.9249	5.207

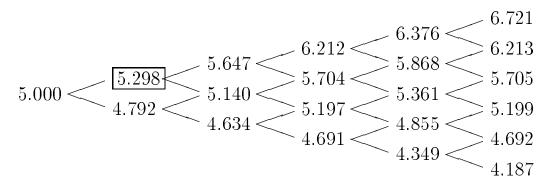
(ie, the spot rates are exactly what we want)

### 10. Options on Eurodollar Futures

- Recall: options on eurodollar futures are like options on 3-month LIBOR
  - we saw this earlier when we examined the "yields" implicit in futures prices
  - there are subtle differences between forward rates and futures that we'll ignore for now
  - 3-month LIBOR ("Y") is related to the 3-month discount factor ("b") by

$$Y = 400 \times (1/b - 1)$$

Since our tree has a 3-month time interval, the tree for Y is easily computed from the one-period discount factors:



\* node (1,1) (box):

"b" = 
$$\exp(-5.264/400) = 0.98693$$
  
"Y" =  $400 \times (1/b - 1) = 5.298$ 

\* not much different from the continuously-compounded short rate, but it reminds us that interest rate conventions are important

- Consider an option with strike K on Y in 3 months
  - the option has cash flows of  $(Y K)^+$ :
  - with K = 5 the cash flows are

- Value of option:
  - \* all-at-once method (multiply cash flows by state prices and add):

$$p = (0.4938)(0.298) = 0.147$$

- Term structure of volatility revisited
  - Objective: compute volatilities for at-the-money options
  - We need forward rates:
    - \* with a 3-month time interval (h = 0.25), 3-month forward rates satisfy

$$1 + F^{n}/400 = b^{n}/b^{n+1} \Rightarrow F^{n} = 400 (b^{n}/b^{n+1} - 1)$$

\* from the discount factors computed above, we get

Maturity Discount Factor Spot Rate Forward Rate

maturriy	Discoult Factor	Spot Rate	roiward itale
0.25	0.9877	4.969	5.045
0.50	0.9754	4.999	5.140
0.75	0.9630	5.030	5.450
1.00	0.9500	5.126	5.360
1.25	0.9375	5.166	5.450
1.50	0.9249	5.207	5.525

(the last one is based on  $b^7$ , which we haven't reported)

- \* Comment: by construction, forward rates are "100 futures prices" (same prices we reported last time)
- Compute prices of at-the-money options (K = F):

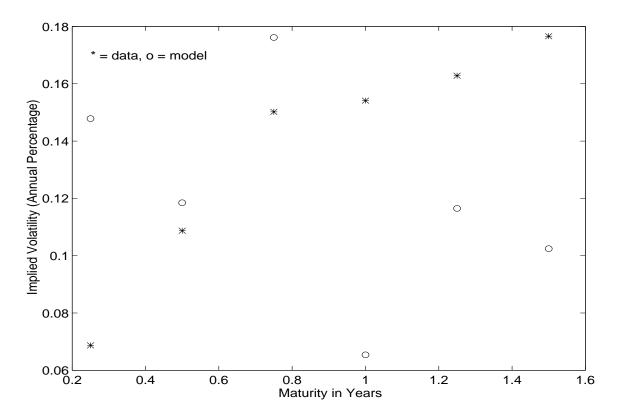
Maturity	Strike	Call Price	Volatility
0.25	5.045	0.1251	0.1478
0.50	5.140	0.1237	0.1185
0.75	5.450	0.1833	0.1762
1.50	5.360	0.1810	0.0654
1.25	5.450	0.2231	0.1165
1.50	5.525	0.2207	0.1025

- Term structure of volatility revisited (continued)
  - How we got these numbers:
    - \* option prices: same approach as above (find cash flows, multiply by state prices, and add); eg,

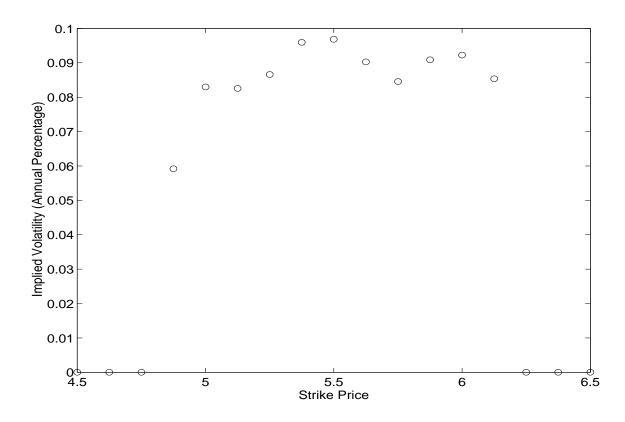
$$0.1251 = (0.4938)(5.298 - 5.045)$$

- \* volatility: inputs are price (above), strike (forward rate), and n-period discount factor (use n-period spot rate) (shortcut: Brenner-Subrahmanyam approximation)
- \* good learning experience: pick a specific maturity and work through all the steps

#### - Result:



- Term structure of volatility revisited (continued)
  - Comments:
    - \* bumpy!
      - · the inevitable result of a discrete model
      - · can be mitigated with smaller time interval
    - \* no obvious pattern to term structure of volatility (if there is one, it's lost in the noise)
    - \* unlike BDT, we can't choose volatilities to fit current term structure of volatility
  - Volatility smile (maturity 9 months):



# Summary

- 1. Many of the most popular fixed income models are based on binomial trees: each period, you go up or down, and (up,down) and (down,up) get you to the same place
- 2. A model consists of a rule for generating interest rates
- 3. Using such a model to value a derivative asset involves the following steps:
  - choose a model
  - choose parameter values
  - compute the asset's cash flows in each node of the tree (this often takes some effort)
  - value the cash flows by either (i) multiplying them by state prices and summing or (ii) computing the value "recursively" (one period at a time, starting at the end)
- 4. Given a model, we can value almost anything with the same technology
- 5. Models differ in their functional form (logs or levels?) and in the flexibility of their parameters (BDT allows input of a volatility term structure, Ho and Lee does not although it could!)
- 6. For options, the discrete set of possibilities of binomial models can lead to "bumpy" prices