The Ritchken-Trevor (RT) Algorithm[a]

- The GARCH model is a continuous-state model.
- To approximate it, we turn to trees with *discrete* states.
- Path dependence in GARCH makes the tree for asset prices explode exponentially (why?).
- We need to mitigate this combinatorial explosion.

[a]Ritchken & Trevor (1999).

The RT Algorithm (continued)

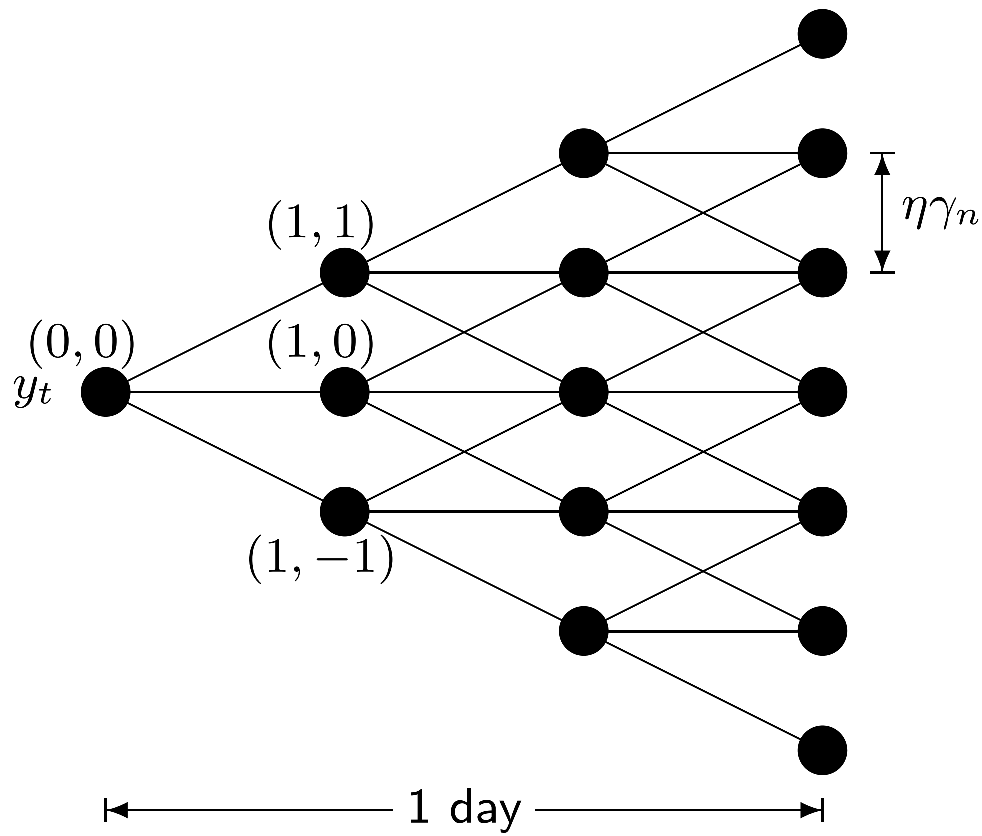
- Partition a day into n periods.
- Three states follow each state (y_t, h_t^2) after a period.
- As the trinomial model combines, each state at date t is followed by $2n+1$ states at date $t+1$ (recall p. 731).
- These $2n+1$ values must approximate the distribution of (y_{t+1}, h_{t+1}^2).
- So the conditional moments (126)–(127) at date $t+1$ on p. 952 must be matched by the trinomial model to guarantee convergence to the continuous-state model.

The RT Algorithm (continued)

- It remains to pick the jump size and the three branching probabilities.
- The role of σ in the Black-Scholes option pricing model is played by h_t in the GARCH model.
- As a jump size proportional to $\sigma/\sqrt{n}$ is picked in the BOPM, a comparable magnitude will be chosen here.
- Define $\gamma \stackrel{\Delta}{=} h_0$, though other multiples of h_0 are possible, and
$$\gamma_n \stackrel{\Delta}{=} \frac{\gamma}{\sqrt{n}}.$$

The RT Algorithm (continued)

- The jump size will be some integer multiple η of γ_n.
- We call η the jump parameter (see next page).
- Obviously, the magnitude of η grows with h_t.
- The middle branch does not change the underlying asset's price.



The seven values on the right approximate the distribution of logarithmic price y_{t+1}.

The RT Algorithm (continued)

- The probabilities for the up, middle, and down branches are

$$p_u = \frac{h_t^2}{2\eta^2\gamma^2} + \frac{r - (h_t^2/2)}{2\eta\gamma\sqrt{n}}, \tag{128}$$

$$p_m = 1 - \frac{h_t^2}{\eta^2\gamma^2}, \tag{129}$$

$$p_d = \frac{h_t^2}{2\eta^2\gamma^2} - \frac{r - (h_t^2/2)}{2\eta\gamma\sqrt{n}}. \tag{130}$$

The RT Algorithm (continued)

- It can be shown that:
  - The trinomial model takes on $2n + 1$ values at date $t + 1$ for y_{t+1}.
  - These values have a matching mean for y_{t+1}.
  - These values have an asymptotically matching variance for y_{t+1}.
- The central limit theorem guarantees convergence as n increases.[a]

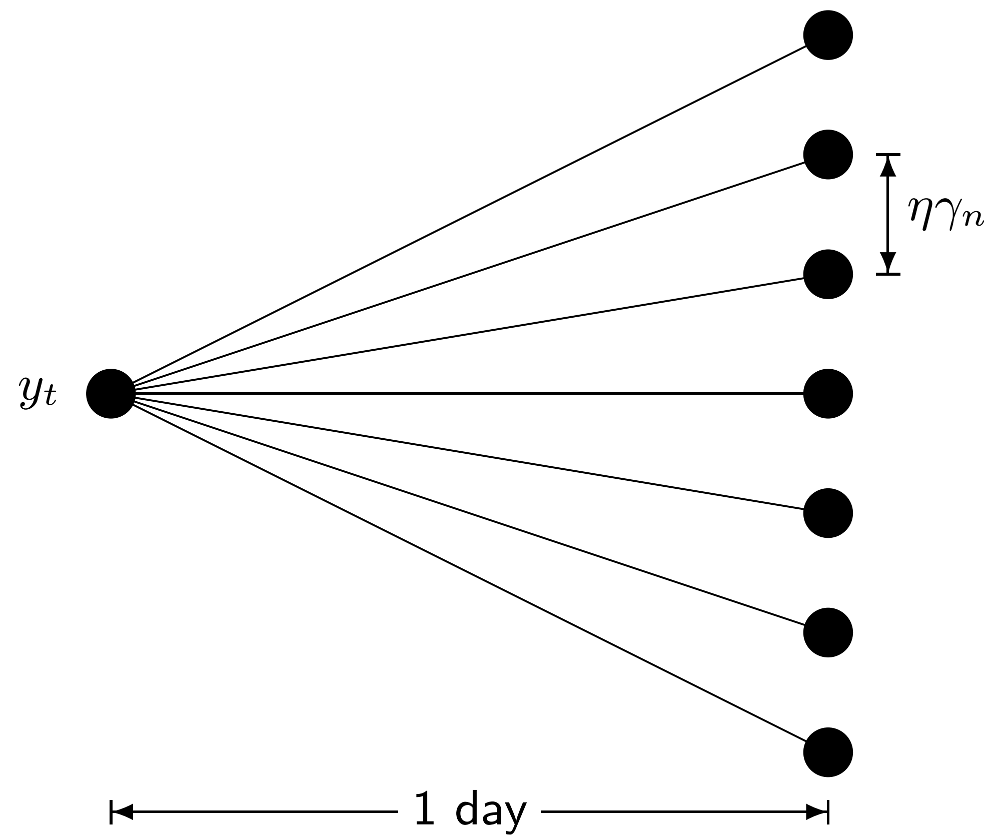
[a] Assume the probabilities are valid.

The RT Algorithm (continued)

- We can dispense with the intermediate nodes *between* dates to create a $(2n+1)$-nomial tree (p. 962).
- The resulting model is multinomial with $2n+1$ branches from any state (y_t, h_t^2).
- There are two reasons behind this manipulation.
  - Interdate nodes are created merely to approximate the continuous-state model after one day.
  - Keeping the interdate nodes results in a tree that can be n times larger.[a]

---

[a] Contrast it with the case on p. 406.



This heptanomial tree is the outcome of the trinomial tree on p. 958 after its intermediate nodes are removed.

The RT Algorithm (continued)

- A node with logarithmic price $y_t + \ell\eta\gamma_n$ at date $t+1$ follows the current node at date t with price y_t, where

$$-n \leq \ell \leq n.$$

- To reach that price in n periods, the number of up moves must exceed that of down moves by exactly ℓ.
- The probability that this happens is

$$P(\ell) \stackrel{\Delta}{=} \sum_{j_u, j_m, j_d} \frac{n!}{j_u!\, j_m!\, j_d!}\, p_u^{j_u} p_m^{j_m} p_d^{j_d},$$

with $j_u, j_m, j_d \geq 0$, $n = j_u + j_m + j_d$, and $\ell = j_u - j_d$.

The RT Algorithm (continued)

- A particularly simple way to calculate the $P(\ell)$s starts by noting that[a]

$$(p_u x + p_m + p_d x^{-1})^n = \sum_{\ell=-n}^{n} P(\ell)\, x^\ell. \tag{131}$$

  - Convince yourself that this trick does the "accounting" correctly.

- So we expand $(p_u x + p_m + p_d x^{-1})^n$ and retrieve the probabilities by reading off the coefficients.

- It can be computed in $O(n^2)$ time, if not less.

[a]C. Wu (R90723065) (2003); Lyuu & C. Wu (R90723065) (2003, 2005).

The RT Algorithm (continued)

- The updating rule (124) on p. 948 must be modified to account for the adoption of the discrete-state model.
- The logarithmic price $y_t + \ell\eta\gamma_n$ at date $t+1$ following state (y_t, h_t^2) is associated with this variance:

$$h_{t+1}^2 = \beta_0 + \beta_1 h_t^2 + \beta_2 h_t^2 (\epsilon'_{t+1} - c)^2, \qquad (132)$$

  – Above, the z-score

$$\epsilon'_{t+1} = \frac{\ell\eta\gamma_n - (r - h_t^2/2)}{h_t}, \quad \ell = 0, \pm 1, \pm 2, \ldots, \pm n,$$

  is a discrete random variable with $2n+1$ values.

The RT Algorithm (continued)

- Different conditional variances h_t^2 may require different η so that the probabilities calculated by Eqs. (128)–(130) on p. 959 lie between 0 and 1.
- This implies varying jump sizes.
- The necessary requirement $p_m \geq 0$ implies $\eta \geq h_t/\gamma$.
- Hence we try

$$\eta = \lceil h_t/\gamma \rceil, \lceil h_t/\gamma \rceil + 1, \lceil h_t/\gamma \rceil + 2, \ldots$$

until valid probabilities are obtained or until their nonexistence is confirmed.

The RT Algorithm (continued)

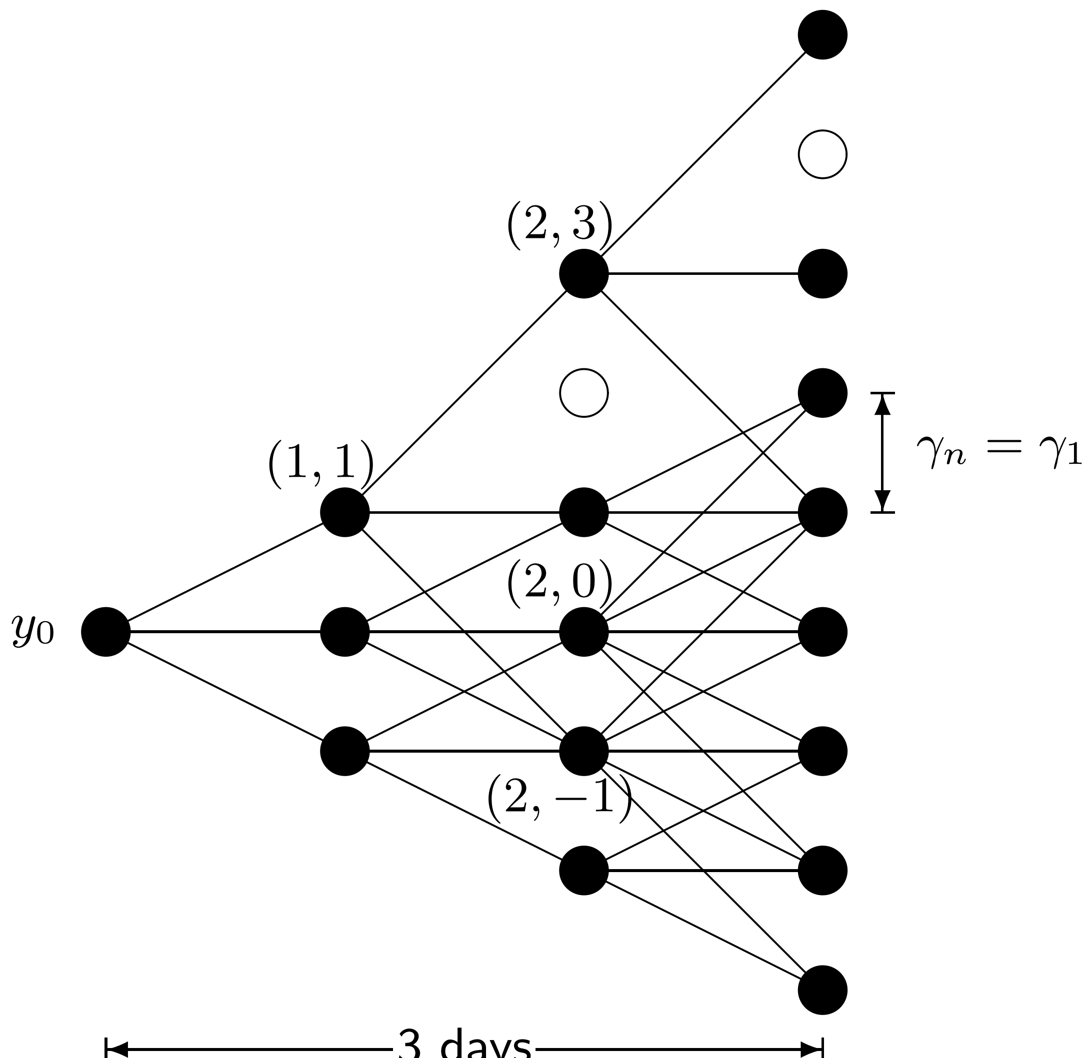
- The sufficient and necessary condition for valid probabilities to exist is[a]

$$\frac{|\, r-(h_t^2/2)\,|}{2\eta\gamma\sqrt{n}} \leq \frac{h_t^2}{2\eta^2\gamma^2} \leq \min\left(1-\frac{|\, r-(h_t^2/2)\,|}{2\eta\gamma\sqrt{n}}, \frac{1}{2}\right).$$

- The plot on p. 968 uses $n = 1$ to illustrate our points for a 3-day model.
- For example, node $(1,1)$ of date 1 and node $(2,3)$ of date 2 pick $\eta = 2$.

---

[a] C. Wu (R90723065) (2003); Lyuu & C. Wu (R90723065) (2003, 2005).


$(2,3)$
$(1,1)$
$\gamma_n = \gamma_1$
y_0
$(2,0)$
$(2,-1)$
3 days

The RT Algorithm (continued)

- The topology of the tree is not a standard combining multinomial tree.
- For example, a few nodes on p. 968 such as nodes $(2, 0)$ and $(2, -1)$ have *multiple* jump sizes.
- The reason is path dependency of the model.
  - Two paths can reach node $(2, 0)$ from the root node, each with a different variance for the node.
  - One variance results in $\eta = 1$.
  - The other results in $\eta = 2$.

The RT Algorithm (concluded)

- The number of possible values of h_t^2 at a node can be exponential.
  - Because each path brings a different variance h_t^2.
- To address this problem, we record only the maximum and minimum h_t^2 at each node.[a]
- Therefore, each node on the tree contains only two states $(y_t, h_{\max}^2)$ and $(y_t, h_{\min}^2)$.
- Each of $(y_t, h_{\max}^2)$ and $(y_t, h_{\min}^2)$ carries its own η and set of $2n + 1$ branching probabilities.

---

[a]Cakici & Topyan (2000). But see p. 1005 for a potential problem.

Negative Aspects of the Ritchken-Trevor Algorithm[a]

- A small n may yield inaccurate option prices.
- But the tree will grow exponentially if n is large enough.
  - Specifically, $n > (1 - \beta_1)/\beta_2$ when $r = c = 0$.
- A large n has another serious problem: The tree cannot grow beyond a certain date.
- Thus the choice of n may be quite limited in practice.
- The RT algorithm can be modified to be free of shortened maturity and exponential complexity.[b]

---

[a] Lyuu & C. Wu (R90723065) (2003, 2005).

[b] Its size is only $O(n^2)$ if $n \leq (\sqrt{(1 - \beta_1)/\beta_2} - c)^2$!

Numerical Examples

- Assume
  - $S_0 = 100$, $y_0 = \ln S_0 = 4.60517$.
  - $r = 0$.
  - $n = 1$.
  - $h_0^2 = 0.0001096$, $\gamma = h_0 = 0.010469$.
  - $\gamma_n = \gamma/\sqrt{n} = 0.010469$.
  - $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, and $c = 0$.

Numerical Examples (continued)

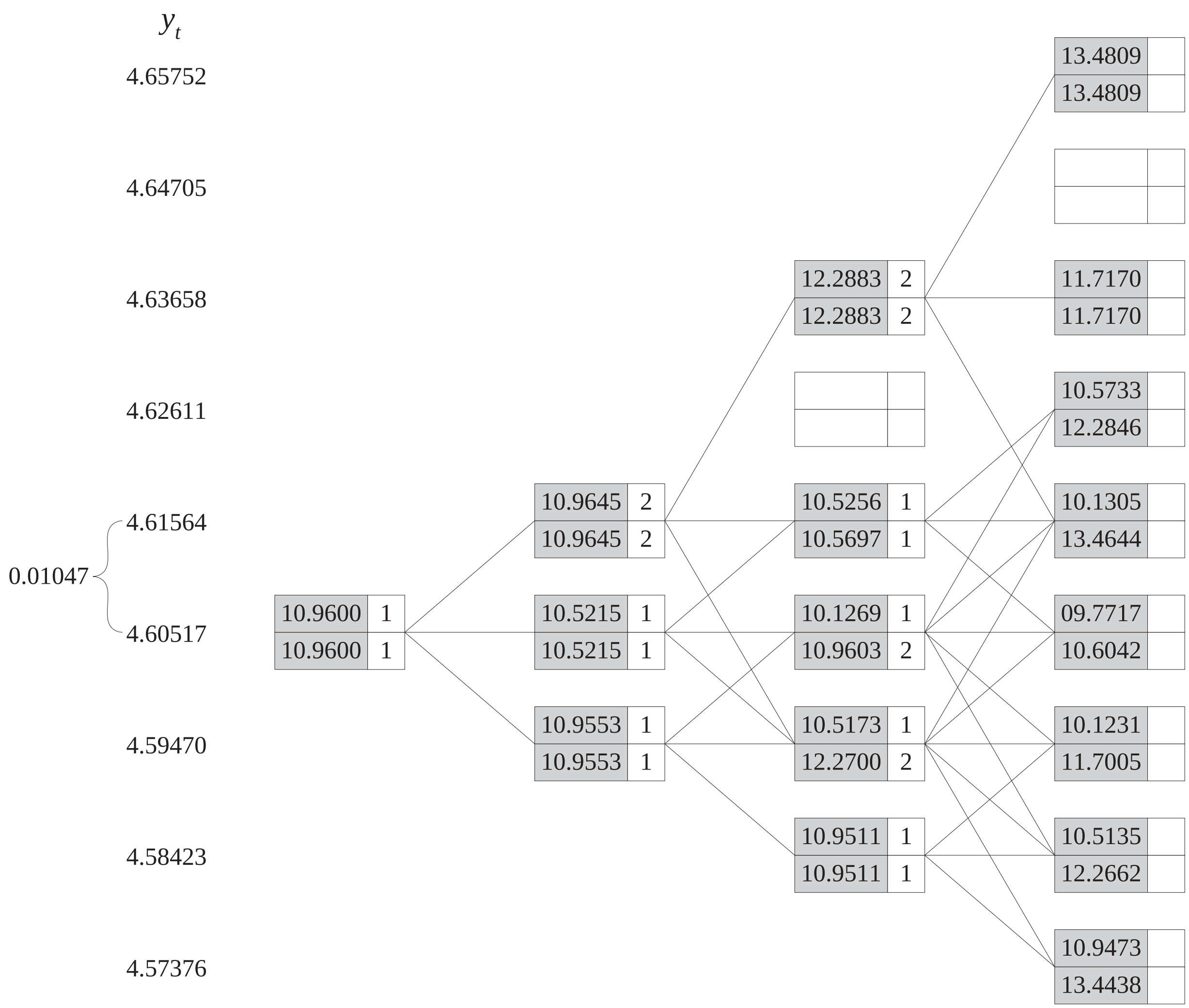
- A daily variance of 0.0001096 corresponds to an annual volatility of

$$\sqrt{365 \times 0.0001096} \approx 20\%.$$

- Let $h^2(i,j)$ denote the variance at node (i,j).
- Initially, $h^2(0,0) = h_0^2 = 0.0001096$.

Numerical Examples (continued)

- Let $h_{\max}^2(i, j)$ denote the maximum variance at node (i, j).
- Let $h_{\min}^2(i, j)$ denote the minimum variance at node (i, j).
- Initially, $h_{\max}^2(0, 0) = h_{\min}^2(0, 0) = h_0^2$.
- The resulting 3-day tree is depicted on p. 975.



y_t
4.65752
4.64705
4.63658
4.62611
4.61564
0.01047
4.60517
4.59470
4.58423
4.57376
10.9600 1
10.9600 1
10.9645 2
10.9645 2
10.5215 1
10.5215 1
10.9553 1
10.9553 1
12.2883 2
12.2883 2
10.5256 1
10.5697 1
10.1269 1
10.9603 2
10.5173 1
12.2700 2
10.9511 1
10.9511 1
13.4809
13.4809
11.7170
11.7170
10.5733
12.2846
10.1305
13.4644
09.7717
10.6042
10.1231
11.7005
10.5135
12.2662
10.9473
13.4438

Numerical Examples (continued)

- A top number inside a gray box refers to the minimum variance $h_{\min}^2$ for the node.
- A bottom number inside a gray box refers to the maximum variance $h_{\max}^2$ for the node.
- Variances are multiplied by 100,000 for readability.
- The top number inside a white box refers to the η for $h_{\min}^2$.
- The bottom number inside a white box refers to the η for $h_{\max}^2$.

Numerical Examples (continued)

- Let us see how the numbers are calculated.
- Start with the root node, node $(0, 0)$.
- Try $\eta = 1$ in Eqs. (128)–(130) on p. 959 first to obtain

$$\begin{aligned} p_u &= 0.4974, \\ p_m &= 0, \\ p_d &= 0.5026. \end{aligned}$$

- As they are valid probabilities, the three branches from the root node use single jumps.

Numerical Examples (continued)

- Move on to node $(1, 1)$.
- It has one predecessor node—node $(0, 0)$—and it takes an up move to reach the current node.
- So apply updating rule (132) on p. 965 with $\ell = 1$ and $h_t^2 = h^2(0, 0)$.
- The result is $h^2(1, 1) = 0.000109645$.

Numerical Examples (continued)

- Because $\lceil h(1,1)/\gamma \rceil = 2$, we try $\eta = 2$ in Eqs. (128)–(130) on p. 959 first to obtain

$$\begin{aligned} p_u &= 0.1237, \\ p_m &= 0.7499, \\ p_d &= 0.1264. \end{aligned}$$

- As they are valid probabilities, the three branches from node $(1,1)$ use double jumps.

Numerical Examples (continued)

- Carry out similar calculations for node $(1, 0)$ with $\ell = 0$ in updating rule (132) on p. 965.
- Carry out similar calculations for node $(1, -1)$ with $\ell = -1$ in updating rule (132).
- Single jump $\eta = 1$ works for both nodes.
- The resulting variances are

$$\begin{aligned} h^2(1,0) &= 0.000105215, \\ h^2(1,-1) &= 0.000109553. \end{aligned}$$

Numerical Examples (continued)

- Node $(2, 0)$ has 2 predecessor nodes, $(1, 0)$ and $(1, -1)$.
- Both have to be considered in deriving the variances.
- Let us start with node $(1, 0)$.
- Because it takes a middle move to reach the current node, we apply updating rule (132) on p. 965 with $\ell = 0$ and $h_t^2 = h^2(1, 0)$.
- The result is $h_{t+1}^2 = 0.000101269$.

Numerical Examples (continued)

- Now move on to the other predecessor node $(1, -1)$.
- Because it takes an up move to reach the current node, apply updating rule (132) on p. 965 with $\ell = 1$ and $h_t^2 = h^2(1, -1)$.
- The result is $h_{t+1}^2 = 0.000109603$.
- We hence record

$$\begin{aligned} h^2_{\min}(2,0) &= 0.000101269, \\ h^2_{\max}(2,0) &= 0.000109603. \end{aligned}$$

Numerical Examples (continued)

- Consider state $h^2_{\max}(2,0)$ first.
- Because $\lceil h_{\max}(2,0)/\gamma \rceil = 2$, we first try $\eta = 2$ in Eqs. (128)–(130) on p. 959 to obtain

$$\begin{aligned} p_u &= 0.1237, \\ p_m &= 0.7500, \\ p_d &= 0.1263. \end{aligned}$$

- As they are valid probabilities, the three branches from node $(2,0)$ with the maximum variance use double jumps.

Numerical Examples (continued)

- Now consider state $h_{\min}^2(2,0)$.
- Because $\lceil h_{\min}(2,0)/\gamma \rceil = 1$, we first try $\eta = 1$ in Eqs. (128)–(130) on p. 959 to obtain

$$
\begin{aligned}
p_u &= 0.4596, \\
p_m &= 0.0760, \\
p_d &= 0.4644.
\end{aligned}
$$

- As they are valid probabilities, the three branches from node $(2, 0)$ with the minimum variance use single jumps.

Numerical Examples (continued)

- Node $(2, -1)$ has 3 predecessor nodes.
- Start with node $(1, 1)$.
- Because it takes *one* down move to reach the current node, we apply updating rule (132) on p. 965 with $\ell = -1$ and $h_t^2 = h^2(1, 1)$.[a]
- The result is $h_{t+1}^2 = 0.0001227$.

[a]Note that it is *not* $\ell = -2$. The reason is that $h(1, 1)$ has $\eta = 2$ (p. 979).

Numerical Examples (continued)

- Now move on to predecessor node $(1, 0)$.
- Because it also takes a down move to reach the current node, we apply updating rule (132) on p. 965 with $\ell = -1$ and $h_t^2 = h^2(1, 0)$.
- The result is $h_{t+1}^2 = 0.000105609$.

Numerical Examples (continued)

- Finally, consider predecessor node $(1, -1)$.
- Because it takes a middle move to reach the current node, we apply updating rule (132) on p. 965 with $\ell = 0$ and $h_t^2 = h^2(1, -1)$.
- The result is $h_{t+1}^2 = 0.000105173$.
- We hence record

$$\begin{aligned} h_{\min}^2(2, -1) &= 0.000105173, \\ h_{\max}^2(2, -1) &= 0.0001227. \end{aligned}$$

Numerical Examples (continued)

- Consider state $h^2_{\max}(2, -1)$.
- Because $\lceil h_{\max}(2, -1)/\gamma \rceil = 2$, we first try $\eta = 2$ in Eqs. (128)–(130) on p. 959 to obtain

$$\begin{aligned} p_u &= 0.1385, \\ p_m &= 0.7201, \\ p_d &= 0.1414. \end{aligned}$$

- As they are valid probabilities, the three branches from node $(2, -1)$ with the maximum variance use double jumps.

Numerical Examples (continued)

- Next, consider state $h^2_{\min}(2, -1)$.
- Because $\lceil h_{\min}(2, -1)/\gamma \rceil = 1$, we first try $\eta = 1$ in Eqs. (128)–(130) on p. 959 to obtain

$$\begin{aligned} p_u &= 0.4773, \\ p_m &= 0.0404, \\ p_d &= 0.4823. \end{aligned}$$

- As they are valid probabilities, the three branches from node $(2, -1)$ with the minimum variance use single jumps.

Numerical Examples (concluded)

- Other nodes at dates 2 and 3 can be handled similarly.
- In general, if a node has k predecessor nodes, then up to $2k$ variances will be calculated using the updating rule.
  - This is because each predecessor node keeps *two* variance numbers.
- But only the maximum and minimum variances will be kept.

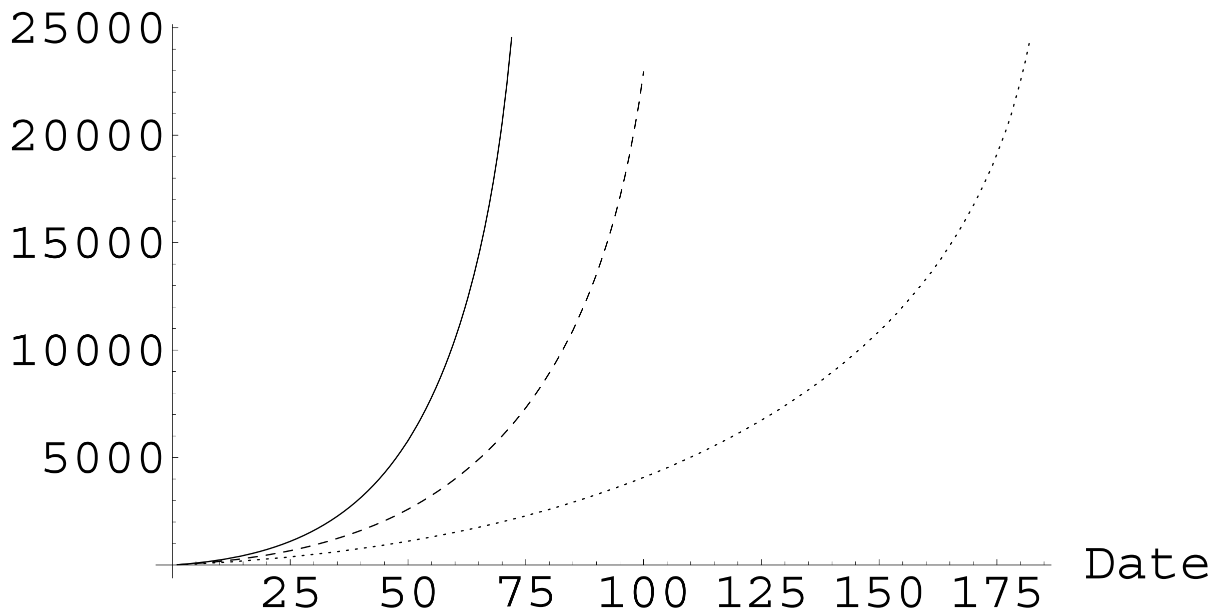
Negative Aspects of the RT Algorithm Revisited[a]

- Recall the problems mentioned on p. 971.
- In our case, combinatorial explosion occurs when

$$n > \frac{1-\beta_1}{\beta_2} = \frac{1-0.9}{0.04} = 2.5$$

  (see the next plot).
- Suppose we are willing to accept the exponential running time and pick $n = 100$ to seek accuracy.
- But the problem of shortened maturity forces the tree to stop at date 9!

[a]Lyuu & C. Wu (R90723065) (2003, 2005).



Dotted line: $n = 3$; dashed line: $n = 4$; solid line: $n = 5$.

Backward Induction on the RT Tree

- After the RT tree is constructed, it can be used to price options by backward induction.
- Recall that each node keeps two variances $h_{\max}^2$ and $h_{\min}^2$.
- We now increase that number to K equally spaced variances between $h_{\max}^2$ and $h_{\min}^2$ at each node.
- Besides the minimum and maximum variances, the other $K-2$ variances in between are linearly interpolated.[a]

[a]In practice, log-linear interpolation works better (Lyuu & C. Wu (R90723065), 2005). Log-cubic interpolation works even better (C. Liu (R92922123), 2005).

Backward Induction on the RT Tree (continued)

- For example, if $K = 3$, then a variance of

$$10.5436 \times 10^{-6}$$

will be added between the maximum and minimum variances at node $(2, 0)$ on p. 975.[a]

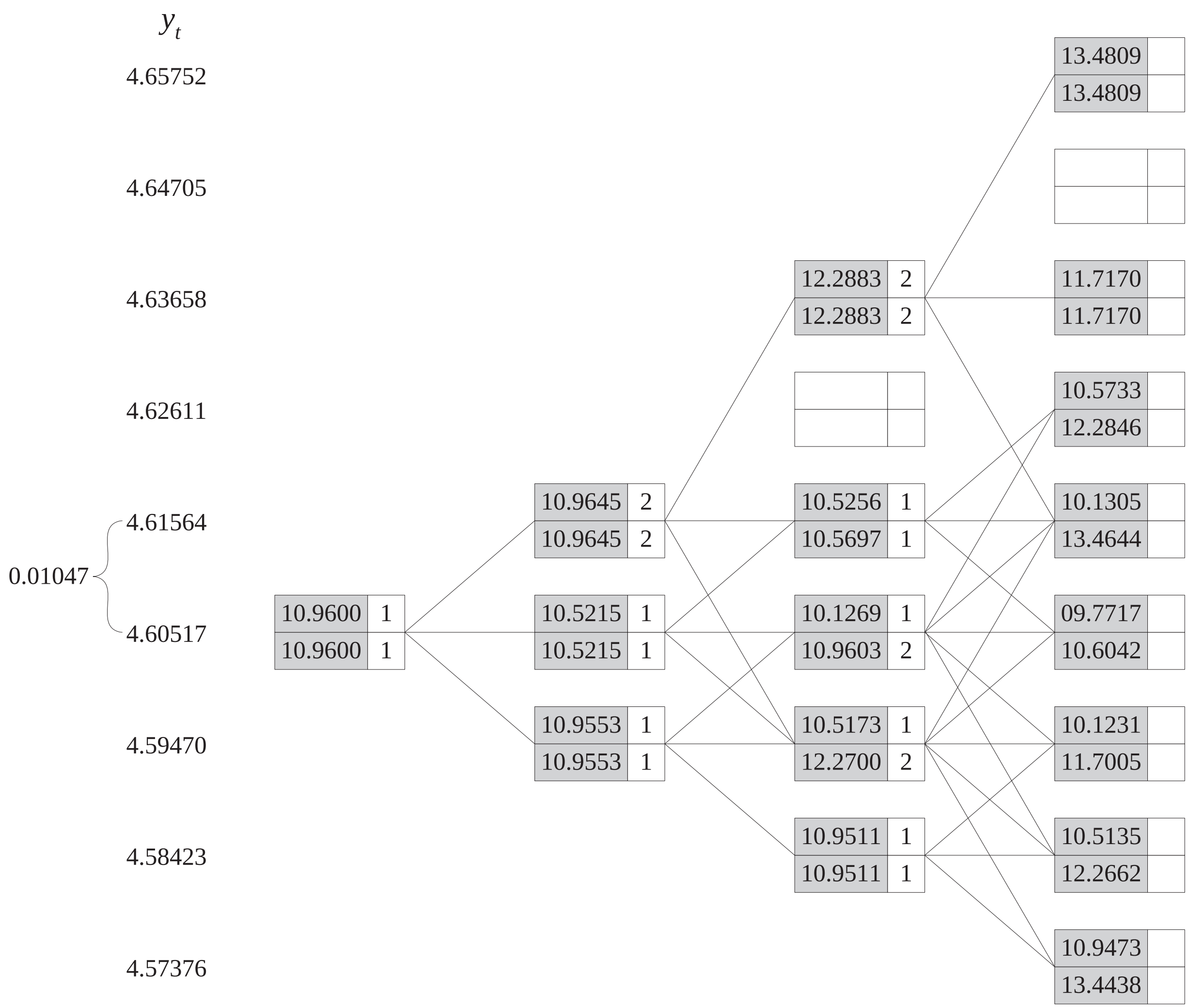
- In general, the kth variance at node (i, j) is

$$h^2_{\min}(i,j) + k\,\frac{h^2_{\max}(i,j) - h^2_{\min}(i,j)}{K-1}, \quad k = 0, 1, \ldots, K-1.$$

- Each interpolated variance's jump parameter and branching probabilities can be computed as before.

---

[a]Repeated on p. 995.

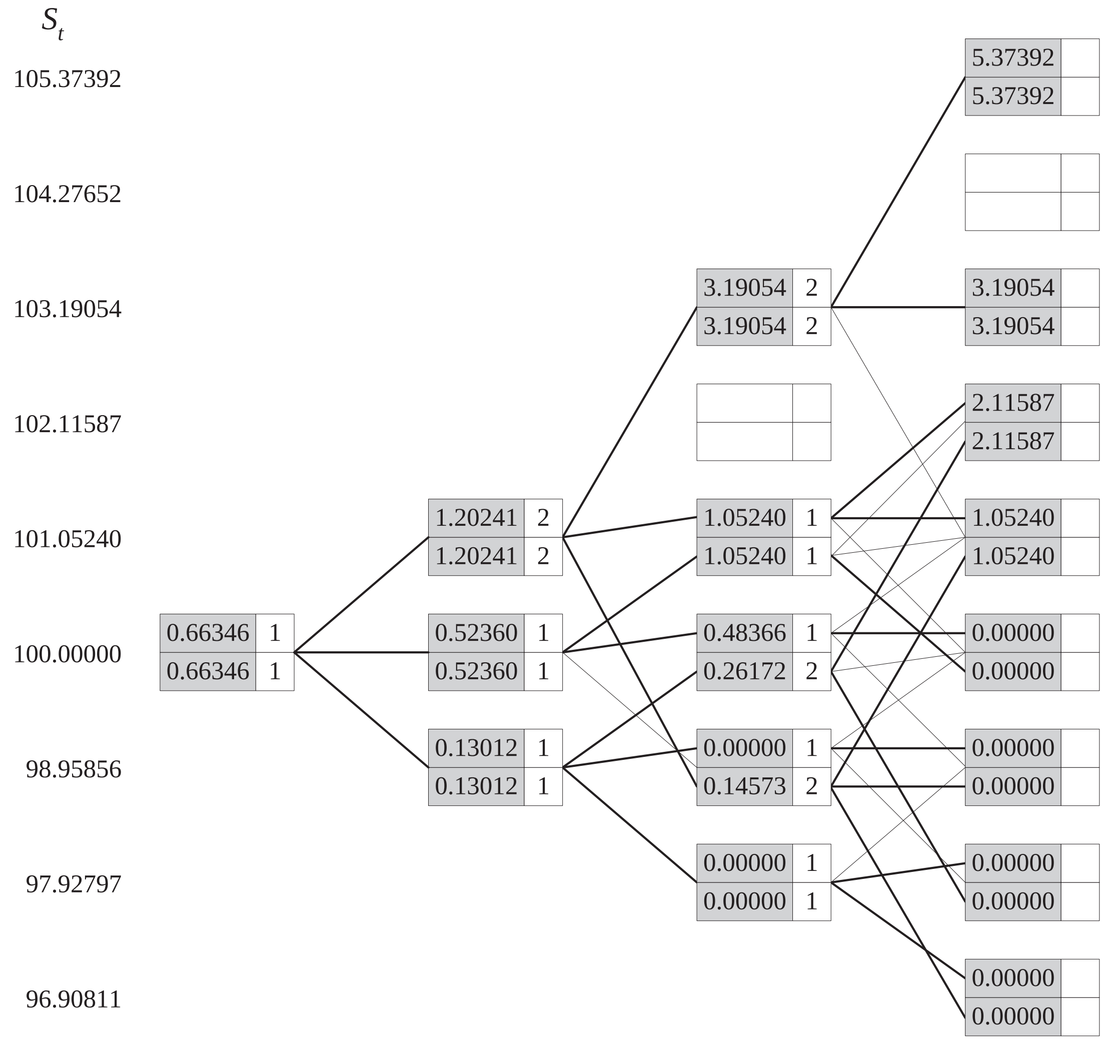

y_t
4.65752
4.64705
4.63658
4.62611
4.61564
0.01047
4.60517
4.59470
4.58423
4.57376
10.9600 1
10.9600 1
10.9645 2
10.9645 2
10.5215 1
10.5215 1
10.9553 1
10.9553 1
12.2883 2
12.2883 2
10.5256 1
10.5697 1
10.1269 1
10.9603 2
10.5173 1
12.2700 2
10.9511 1
10.9511 1
13.4809
13.4809
11.7170
11.7170
10.5733
12.2846
10.1305
13.4644
09.7717
10.6042
10.1231
11.7005
10.5135
12.2662
10.9473
13.4438

Backward Induction on the RT Tree (concluded)

- Suppose a variance falls between two of the K variances during backward induction.
- Linear interpolation of the option prices corresponding to the two bracketing variances will be used as the approximate option price.
- The above ideas are reminiscent of the ones on p. 449, where we dealt with Asian options.

Numerical Examples

- We next use the tree on p. 995 to price a European call option with a strike price of 100 and expiring at date 3.
- Recall that the riskless interest rate is zero.
- Assume $K = 2$; hence there are no interpolated variances.
- The pricing tree is shown on p. 998 with a call price of 0.66346.
  - The branching probabilities needed in backward induction can be found on p. 999.



S_t
105.37392
104.27652
103.19054
102.11587
101.05240
100.00000
98.95856
97.92797
96.90811
0.66346 1
0.66346 1
1.20241 2
1.20241 2
0.52360 1
0.52360 1
0.13012 1
0.13012 1
3.19054 2
3.19054 2
1.05240 1
1.05240 1
0.48366 1
0.26172 2
0.00000 1
0.14573 2
0.00000 1
0.00000 1
5.37392
5.37392
3.19054
3.19054
2.11587
2.11587
1.05240
1.05240
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000

rb[*i*][0] / rb[*i*][1]

	rb[0][ ]	rb[1][ ]	rb[2][ ]	rb[3][ ]
rb[*i*][0]	0	−1	−2	−3
rb[*i*][1]	0	1	3	5

$h^2[i][j][0]$ / $h^2[i][j][1]$

j	h^2[0][ ][ ]	h^2[1][ ][ ]	h^2[2][ ][ ]	h^2[3][ ][ ]
5				13.4809 13.4809
4				
3			12.2883 12.2883	11.7170 11.7170
2				10.5733 12.2846
1		10.9645 10.9645	10.5256 10.5697	10.1305 13.4644
0	10.9600 10.9600	10.5215 10.5215	10.1269 10.9603	09.7717 10.6042
−1		10.9553 10.9553	10.5173 12.2700	10.1231 11.7005
−2			10.9511 10.9511	10.5135 12.2662
−3				10.9473 13.4438

η[*i*][*j*][0] / η[*i*][*j*][1]

j	η[0][ ][ ]	η[1][ ][ ]	η[2][ ][ ]
3			2 2
2			
1		2 2	1 1
0	1 1	1 1	1 2
−1		1 1	1 2
−2			1 1

p[*i*][*j*][0][1] *p*[*i*][*j*][1][1]
p[*i*][*j*][0][0] *p*[*i*][*j*][1][0]
p[*i*][*j*][0][−1] *p*[*i*][*j*][1][−1]

j	*p*[0][ ][ ][ ]	*p*[1][ ][ ][ ]	*p*[2][ ][ ][ ]
3			0.1387 0.1387 0.7197 0.7197 0.1416 0.1416
2			
1		0.1237 0.1237 0.7499 0.7499 0.1264 0.1264	0.4777 0.4797 0.0396 0.0356 0.4827 0.4847
0	0.4974 0.4974 0.0000 0.0000 0.5026 0.5026	0.4775 0.4775 0.0400 0.0400 0.4825 0.4825	0.4596 0.1237 0.0760 0.7500 0.4644 0.1263
−1		0.4972 0.4972 0.0004 0.0004 0.5024 0.5024	0.4773 0.1385 0.0404 0.7201 0.4823 0.1414
−2			0.4970 0.4970 0.0008 0.0008 0.5022 0.5022

Numerical Examples (continued)

- Let us derive some of the numbers on p. 998.
- A gray line means the updated variance falls strictly between $h^2_{\max}$ and $h^2_{\min}$.
- The option price for a terminal node at date 3 equals $\max(S_3 - 100, 0)$, independent of the variance level.
- Now move on to nodes at date 2.
- The option price at node $(2, 3)$ depends on those at nodes $(3, 5)$, $(3, 3)$, and $(3, 1)$.
- It therefore equals

$$0.1387 \times 5.37392 + 0.7197 \times 3.19054 + 0.1416 \times 1.05240 = 3.19054.$$

Numerical Examples (continued)

- Option prices for other nodes at date 2 can be computed similarly.
- For node $(1, 1)$, the option price for both variances is

$$0.1237 \times 3.19054 + 0.7499 \times 1.05240 + 0.1264 \times 0.14573 = 1.20241.$$

- Node $(1, 0)$ is most interesting.
- We knew that a down move from it gives a variance of 0.000105609.
- This number falls between the minimum variance 0.000105173 and the maximum variance 0.0001227 at node $(2, -1)$ on p. 999.

Numerical Examples (continued)

- The option price corresponding to the minimum variance is 0 (p. 999).
- The option price corresponding to the maximum variance is 0.14573.
- The equation

$$x \times 0.000105173 + (1 - x) \times 0.0001227 = 0.000105609$$

is satisfied by $x = 0.9751$.

- So the option for the down state is approximated by

$$x \times 0 + (1 - x) \times 0.14573 = 0.00362.$$

Numerical Examples (continued)

- The up move leads to the state with option price 1.05240.
- The middle move leads to the state with option price 0.48366.
- The option price at node $(1, 0)$ is finally calculated as

$$0.4775 \times 1.05240 + 0.0400 \times 0.48366 + 0.4825 \times 0.00362 = 0.52360.$$

Numerical Examples (continued)

- A variance following an interpolated variance may exceed the maximum variance or be exceeded by the minimum variance.
- When this happens, the option price corresponding to the maximum or minimum variance will be used during backward induction.[a]
- This act tends to reduce the dynamic range of the variance.

---

[a] Cakici & Topyan (2000).

Numerical Examples (concluded)

- Worse, an interpolated variance may choose a branch that goes into a node that is *not* reached in forward induction.[a]
- In this case, the algorithm fails.
- The RT algorithm does not have this problem.
  - This is because all interpolated variances are involved in the forward-induction phase.
- It may be hard to calculate the implied β_1 and β_2 from option prices.[b]

---

[a] Lyuu & C. Wu (R90723065) (2005).
[b] Y. Chang (B89704039, R93922034) (2006).

Complexities of GARCH Models[a]

- The RT algorithm explodes exponentially if n is big enough (p. 971).
- The mean-tracking tree of Lyuu and Wu (2005) makes sure explosion does not happen if n is not too large.[b]
- The next page summarizes the situations for many GARCH option pricing models.
  - Our earlier treatment is for NGARCH only.

[a]Lyuu & C. Wu (R90723065) (2003, 2005).
[b]Similar to, but earlier than, the binomial-trinomial tree on pp. 754ff.

Complexities of GARCH Models (concluded)[a]

Model	Explosion	Non-explosion
NGARCH	$\beta_1 + \beta_2 n > 1$	$\beta_1 + \beta_2(\sqrt{n} + \lambda + c)^2 \leq 1$
LGARCH	$\beta_1 + \beta_2 n > 1$	$\beta_1 + \beta_2(\sqrt{n} + \lambda)^2 \leq 1$
AGARCH	$\beta_1 + \beta_2 n > 1$	$\beta_1 + \beta_2(\sqrt{n} + \lambda)^2 \leq 1$
GJR-GARCH	$\beta_1 + \beta_2 n > 1$	$\beta_1 + (\beta_2 + \beta_3)(\sqrt{n} + \lambda)^2 \leq 1$
TS-GARCH	$\beta_1 + \beta_2 \sqrt{n} > 1$	$\beta_1 + \beta_2(\lambda + \sqrt{n}) \leq 1$
TGARCH	$\beta_1 + \beta_2 \sqrt{n} > 1$	$\beta_1 + (\beta_2 + \beta_3)(\lambda + \sqrt{n}) \leq 1$
Heston-Nandi	$\beta_1 + \beta_2(c - \frac{1}{2})^2 > 1$ & $c \leq \frac{1}{2}$	$\beta_1 + \beta_2 c^2 \leq 1$
VGARCH	$\beta_1 + (\beta_2/4) > 1$	$\beta_1 \leq 1$

[a] Y. C. Chen (R95723051) (2008); Y. C. Chen (R95723051), Lyuu, & Wen (D94922003) (2012).

Introduction to Term Structure Modeling

The fox often ran to the hole
by which they had come in,
to find out if his body was still thin enough
to slip through it.
— *Grimm's Fairy Tales*

And the worst thing you can have
is models and spreadsheets.
— Warren Buffet (2008, May 3)

Renaissance is 100% model driven.[a]
James Simons (2015, May 13, 37:09)

[a] https://www.youtube.com/watch?v=QNznD9hMEh0

Outline

- Use the binomial interest rate tree to model stochastic term structure.
  - Illustrates the basic ideas underlying future models.
  - Applications are generic in that pricing and hedging methodologies can be easily adapted to other models.
- Although the idea is similar to the earlier one used in option pricing, the current task is more complicated.
  - The evolution of an entire term structure, not just a single stock price, is to be modeled.
  - Interest rates of various maturities cannot evolve arbitrarily, or arbitrage profits may occur.

Issues

- A stochastic interest rate model performs two tasks.
  - Provides a stochastic process that defines future term structures without arbitrage profits.
  - "Consistent" with the observed term structures.

History

- The methodology was founded by Merton (1970).
- Modern interest rate modeling is often traced to 1977 when Vasicek and Cox, Ingersoll, and Ross developed simultaneously their influential models.
- Early models have fitting problems because they may not price today's benchmark bonds correctly.
- An alternative approach pioneered by Ho and Lee (1986) makes fitting the market yield curve mandatory.
- Models based on such a paradigm are called (somewhat misleadingly) arbitrage-free or no-arbitrage models.

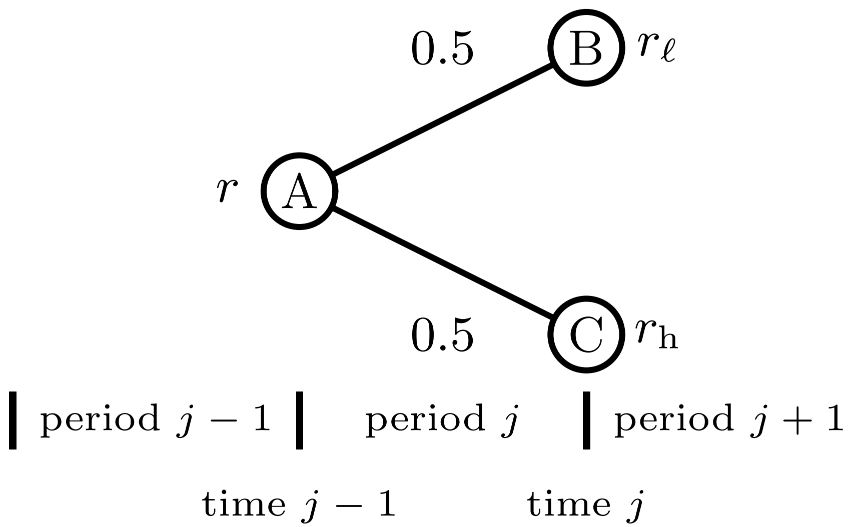
Binomial Interest Rate Tree

- Goal is to construct a no-arbitrage interest rate tree consistent with the yields and/or yield volatilities of zero-coupon bonds of all maturities.
  - This procedure is called calibration.[a]
- Pick a binomial tree model in which the logarithm of the future short rate obeys the binomial distribution.
  - Like the CRR tree for pricing options.
- The limiting distribution of the short rate at any future time is hence lognormal.

[a] Derman (2004), "complexity without calibration is pointless."

Binomial Interest Rate Tree (continued)

- A binomial tree of future short rates is constructed.
- Every short rate is followed by two short rates in the following period (p. 1016).
- In the figure on p. 1016, 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 $j+1$.


0.5
B
r_ℓ
r
A
0.5
C
r_{h}
period $j-1$
period j
period $j+1$
time $j-1$
time j

Binomial Interest Rate Tree (continued)

- This may take one of two possible values:
  - r_ℓ: the "low" short-rate outcome at node B.
  - $r_{\rm h}$: 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.
- This model is attributed to Salomon Brothers.[a]

[a]Tuckman (2002).

Binomial Interest Rate Tree (continued)

- 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.
- Hence the volatility of $\ln r$ after Δt time is[a]

$$\sigma = \frac{1}{2} \frac{1}{\sqrt{\Delta t}} \ln \left(\frac{r_{\rm h}}{r_\ell} \right). \tag{133}$$

- Above, σ is annualized, whereas r_ℓ and $r_{\rm h}$ are period based.

[a]See Exercise 23.2.3 in text.

Binomial Interest Rate Tree (continued)

- Note that
$$\frac{r_{\mathrm{h}}}{r_\ell} = e^{2\sigma\sqrt{\Delta t}}.$$
- Thus greater volatility, hence uncertainty, leads to larger r_{h}/r_ℓ and wider ranges of possible short rates.
- The ratio r_{h}/r_ℓ may depend on time if the volatility is a function of time.
- Note that r_{h}/r_ℓ has nothing to do with the current short rate r if σ is independent of r.

Binomial Interest Rate Tree (continued)

- In general there are j possible rates[a] for *period* j,

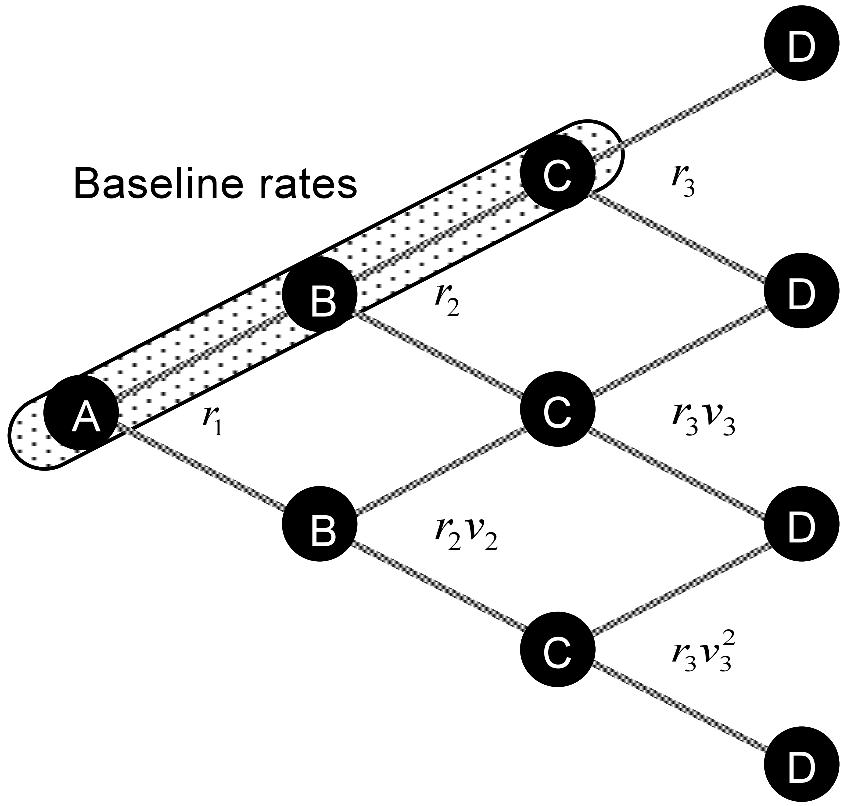
  $$r_j,\ r_j v_j,\ r_j v_j^2,\ \ldots,\ r_j v_j^{j-1},$$

  where

  $$v_j \stackrel{\Delta}{=} e^{2\sigma_j \sqrt{\Delta t}} \tag{134}$$

  is the multiplicative ratio for the rates in period j (see figure on next page).
- We shall call r_j the baseline rates.
- The subscript j in σ_j above is meant to emphasize that the short rate volatility may be time dependent.

[a]Not $j + 1$.


Baseline rates
A
B
B
C
C
C
D
D
D
D
r_1
r_2
$r_2 v_2$
r_3
$r_3 v_3$
$r_3 v_3^2$

Binomial Interest Rate Tree (concluded)

- In the limit, the short rate follows

$$r(t) = \mu(t)\, e^{\sigma(t)\, W(t)}. \tag{135}$$

  - The (percent) short rate volatility $\sigma(t)$ is a deterministic function of time.

- The expected value of $r(t)$ equals $\mu(t)\, e^{\sigma(t)^2 (t/2)}$.
- Hence 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 to some long-term mean.

Memory Issues

- Path independency: The term structure at any node is independent of the path taken to reach it.
- So only the baseline rates r_i and the multiplicative ratios v_i need to be stored in computer memory.
- This takes up only $O(n)$ space.[a]
- Storing the whole tree would take up $O(n^2)$ space.
  - Daily interest rate movements for 30 years require roughly $(30 \times 365)^2/2 \approx 6 \times 10^7$ double-precision floating-point numbers (half a gigabyte!).

[a]Throughout, n denotes the depth of the tree.

Set Things in Motion

- The abstract process is now in place.
- We need the yields to maturities of the riskless bonds that make up the benchmark yield curve and their volatilities.
- In the U.S., for example, the on-the-run yield curve obtained by the most recently issued Treasury securities may be used as the benchmark curve.

Set Things in Motion (concluded)

- The term structure of (yield) volatilities[a] can be estimated from:
  - Historical data (historical volatility).
  - Or interest rate option prices such as cap prices (implied volatility).
- The binomial tree should be found that is consistent with both term structures.
- Here we focus on the term structure of interest rates.

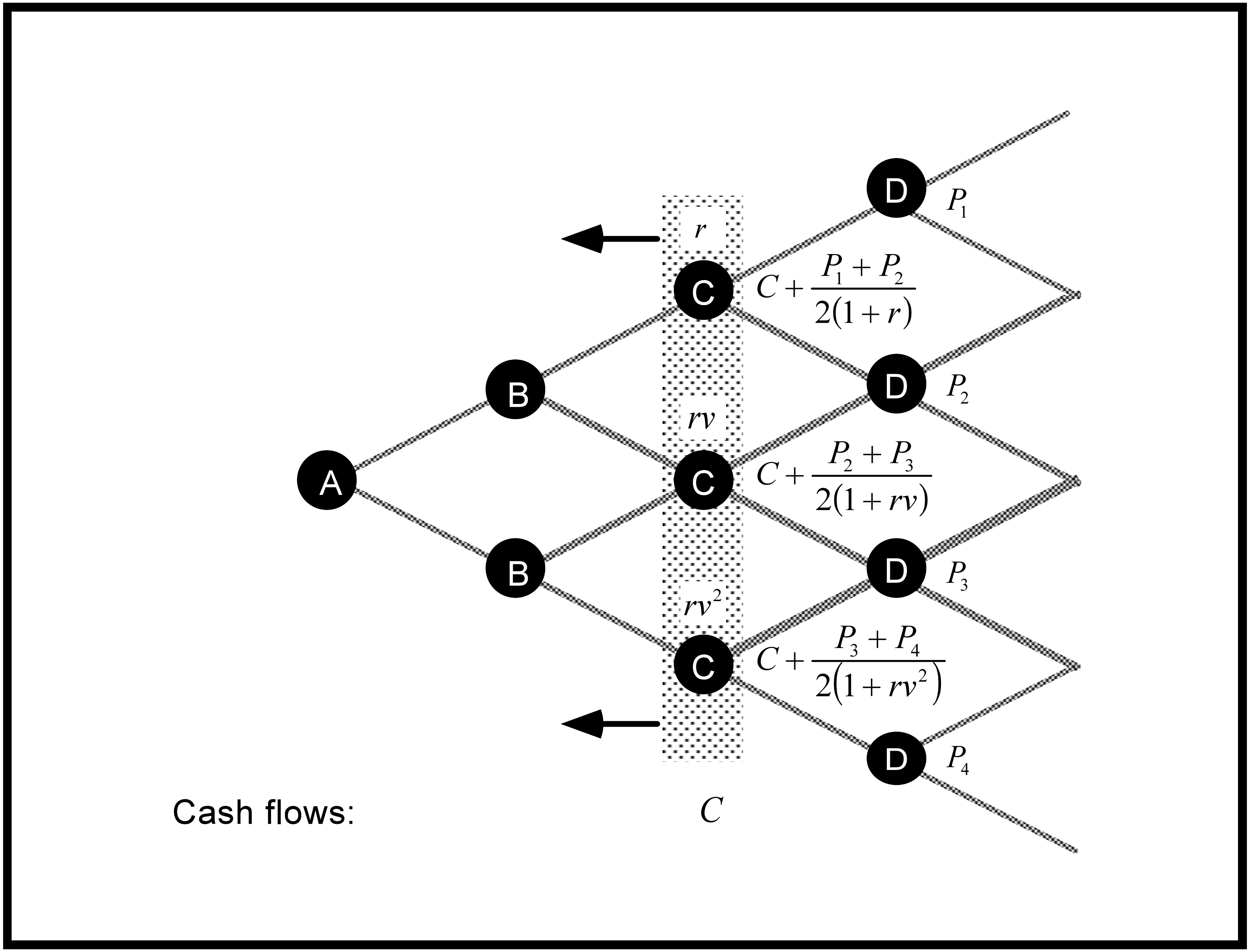
[a]Or simply the volatility (term) structure.

Model Term Structures

- The model price is computed by backward induction.
- Refer back to the figure on p. 1016.
- Given that the values at nodes B and C are P_{B} and P_{C}, respectively, the value at node A is then

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

- We compute the values column by column (see next page).
- This takes $O(n^2)$ time and $O(n)$ space.



D
P_1
r
C
$C + \frac{P_1 + P_2}{2(1+r)}$
D
P_2
B
rv
A
C
$C + \frac{P_2 + P_3}{2(1+rv)}$
B
D
P_3
rv^2
C
$C + \frac{P_3 + P_4}{2(1+rv^2)}$
D
P_4
Cash flows:
C

Term Structure Dynamics

- An n-period zero-coupon bond's price can be computed by assigning \$1 to every node at period n and then applying backward induction.
- Repeating this step for $n = 1, 2, \ldots$, one obtains the market discount function implied by the tree.
- The tree therefore determines a term structure.
- It also contains a term structure dynamics.
  - Taking any node in the tree as the current state induces a binomial interest rate tree and, again, a term structure.

Sample Term Structure

- We shall construct interest rate trees consistent with the sample term structure in the following table.
  - This is calibration (the reverse of pricing).
- Assume the short rate volatility is such that

$$v \stackrel{\Delta}{=} \frac{r_{\rm h}}{r_\ell} = 1.5,$$

independent of time.

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

An Approximate Calibration Scheme

- Start with the implied one-period forward rates.
- Equate the expected short rate with the forward rate.[a]
- 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 via[b]

$$f_j = \frac{d(j)}{d(j+1)} - 1.$$

[a] See Exercise 5.6.6 in text.

[b] See Exercise 5.6.3 in text.

An Approximate Calibration Scheme (continued)

- Since the ith short rate $r_j v_j^{i-1}$, $1 \leq i \leq j$, occurs with probability $2^{-(j-1)} \binom{j-1}{i-1}$, this means

$$\sum_{i=1}^{j} 2^{-(j-1)} \binom{j-1}{i-1} r_j v_j^{i-1} = f_j.$$

- Thus

$$r_j = \left(\frac{2}{1+v_j} \right)^{j-1} f_j. \tag{136}$$

- This binomial interest rate tree is trivial to set up (implicitly), in $O(n)$ time.

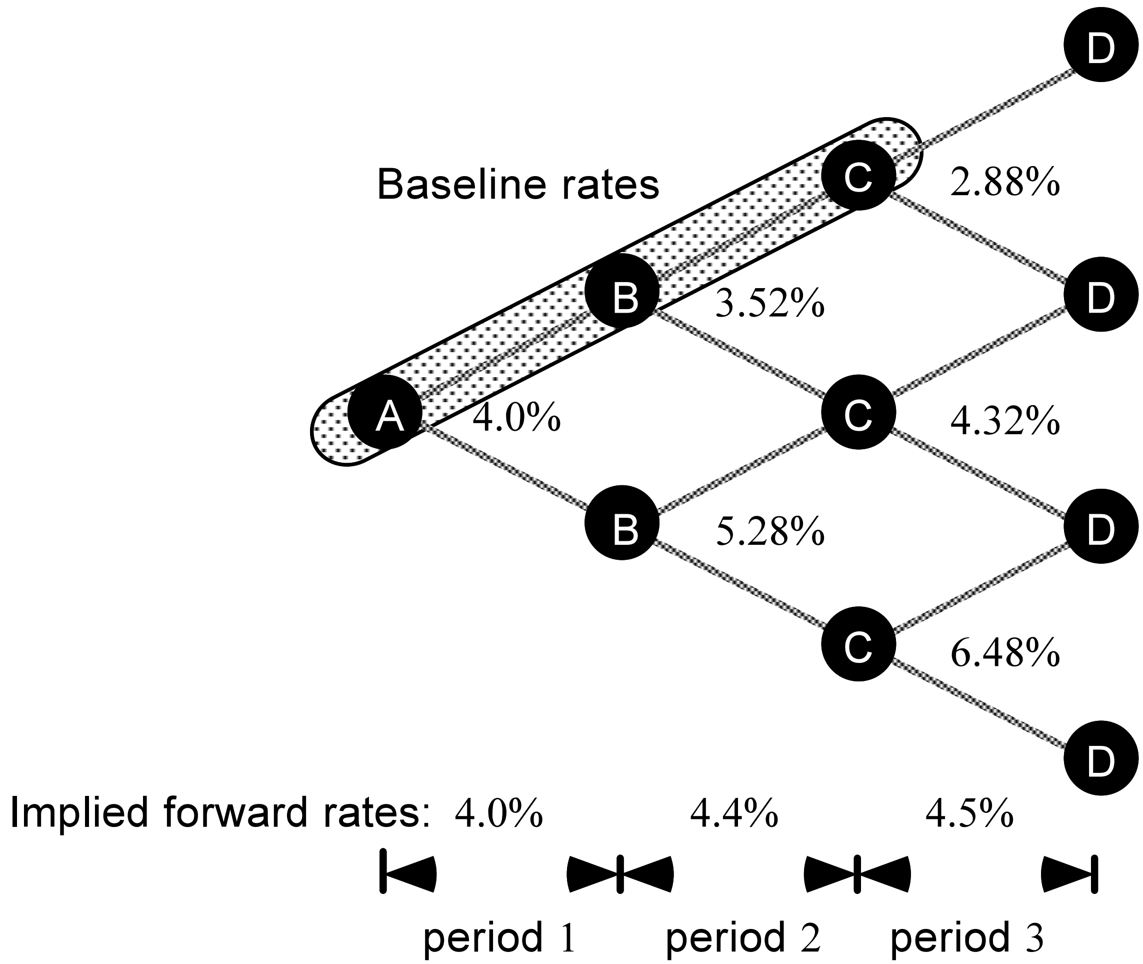
An Approximate Calibration Scheme (continued)

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

$$\frac{1}{4} \times \frac{1}{1.04} \times \left(\frac{1}{1.0352} \times \left(\frac{1}{1.0288} + \frac{1}{1.0432} \right) + \frac{1}{1.0528} \times \left(\frac{1}{1.0432} + \frac{1}{1.0648} \right) \right)$$

  or 0.88155, which exceeds discount factor 0.88135.

- The tree is thus *not* calibrated.



Baseline rates
D
C
2.88%
B
3.52%
D
A
4.0%
C
4.32%
B
5.28%
D
C
6.48%
D
Implied forward rates: 4.0%
4.4%
4.5%
period 1
period 2
period 3

An Approximate Calibration Scheme (concluded)

- Indeed, this bias is inherent: The tree *overprices* the bonds.[a]
- Suppose we replace the baseline rates r_j by $r_j v_j$.
- Then the resulting tree *underprices* the bonds.[b]
- The true baseline rates are thus bounded between r_j and $r_j v_j$.

[a] See Exercise 23.2.4 in text.
[b] Lyuu & C. Wang (F95922018) (2009, 2011).

Issues in Calibration

- The model prices generated by the binomial interest rate tree should match the observed market prices.
- Perhaps the most crucial aspect of model building.
- Treat the backward induction for the model price of the m-period zero-coupon bond as computing some function $f(r_m)$ of the unknown baseline rate r_m for period m.
- A root-finding method is applied to solve $f(r_m) = P$ for r_m given the zero's price P and $r_1, r_2, \ldots, r_{m-1}$.
- This procedure is carried out for $m = 1, 2, \ldots, n$.
- It runs in $O(n^3)$ time.

Binomial Interest Rate Tree Calibration

- Calibration can be accomplished in $O(n^2)$ time by the use of forward induction.[a]
- The scheme records how much \$1 at a node contributes to the model price.
- This number is called the state price (p. 212), the Arrow-Debreu price, or Green's function.
  - It is the price of a state contingent claim that pays \$1 at that particular node (state) and 0 elsewhere.
- The column of state prices will be established by moving *forward* from time 0 to time n.

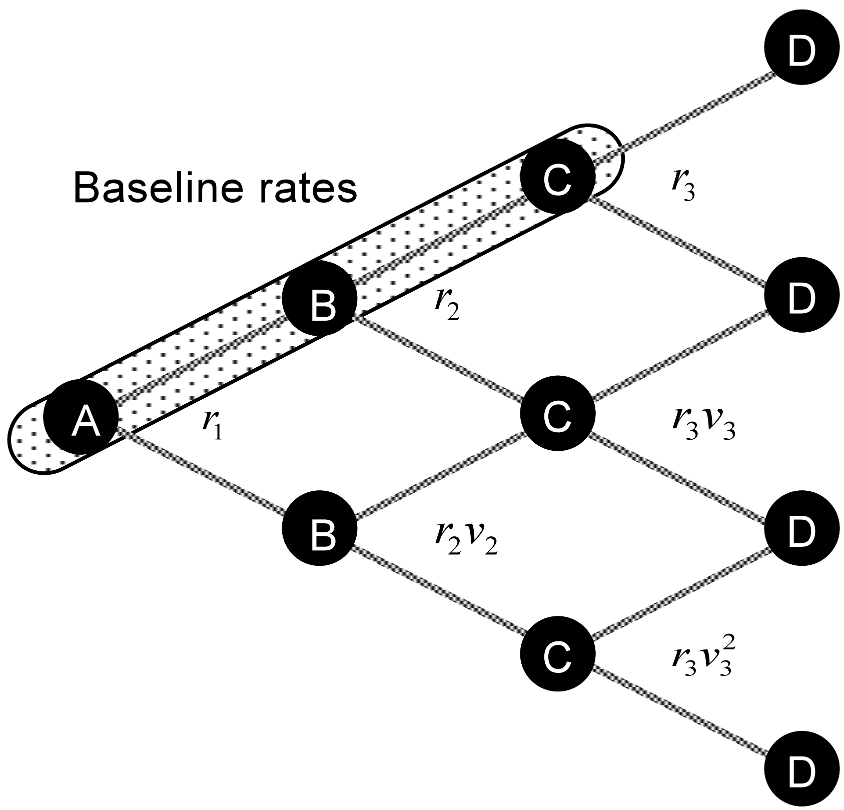
[a] Jamshidian (1991).

Binomial Interest Rate Tree Calibration (continued)

- Suppose we are at *time* j and there are $j+1$ nodes.
  - The unknown baseline rate for *period* j is $r \stackrel{\Delta}{=} r_j$.
  - The multiplicative ratio is $v \stackrel{\Delta}{=} v_j$.
  - $P_1, P_2, \ldots, P_j$ are the known state prices at *earlier* time $j-1$.
  - They have rates $r, rv, \ldots, rv^{j-1}$ for period j.[a]
- By definition, $\sum_{i=1}^{j} P_i$ is the price of the $(j-1)$-period zero-coupon bond.
- We want to find r based on $P_1, P_2, \ldots, P_j$ and the price of the j-period zero-coupon bond.

[a] Recall p. 1021, repeated on next page.

Binomial Interest Rate Tree Calibration (continued)



Binomial Interest Rate Tree Calibration (continued)

- 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 present value of

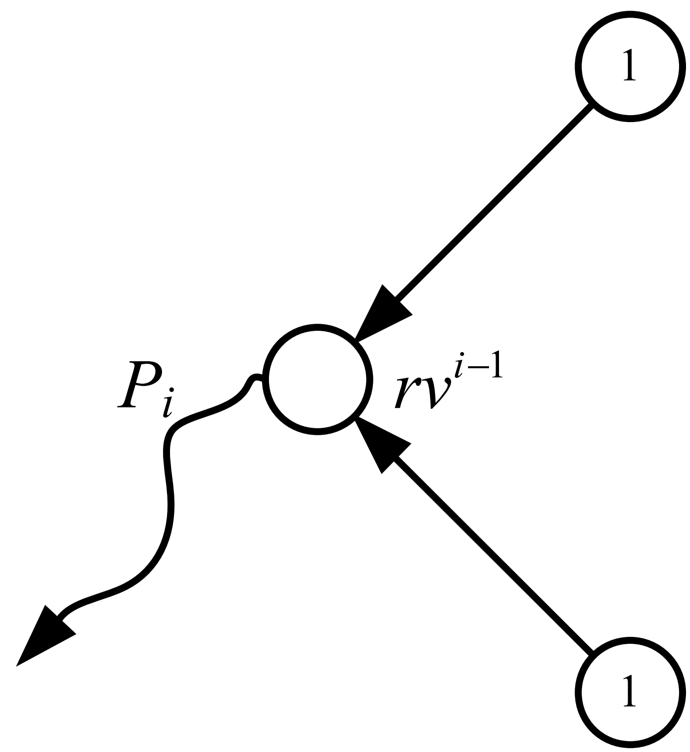
$$g(r) \triangleq \frac{P_1}{(1+r)} + \frac{P_2}{(1+rv)} + \frac{P_3}{(1+rv^2)} + \cdots + \frac{P_j}{(1+rv^{j-1})}$$

(see next plot).

- So we solve

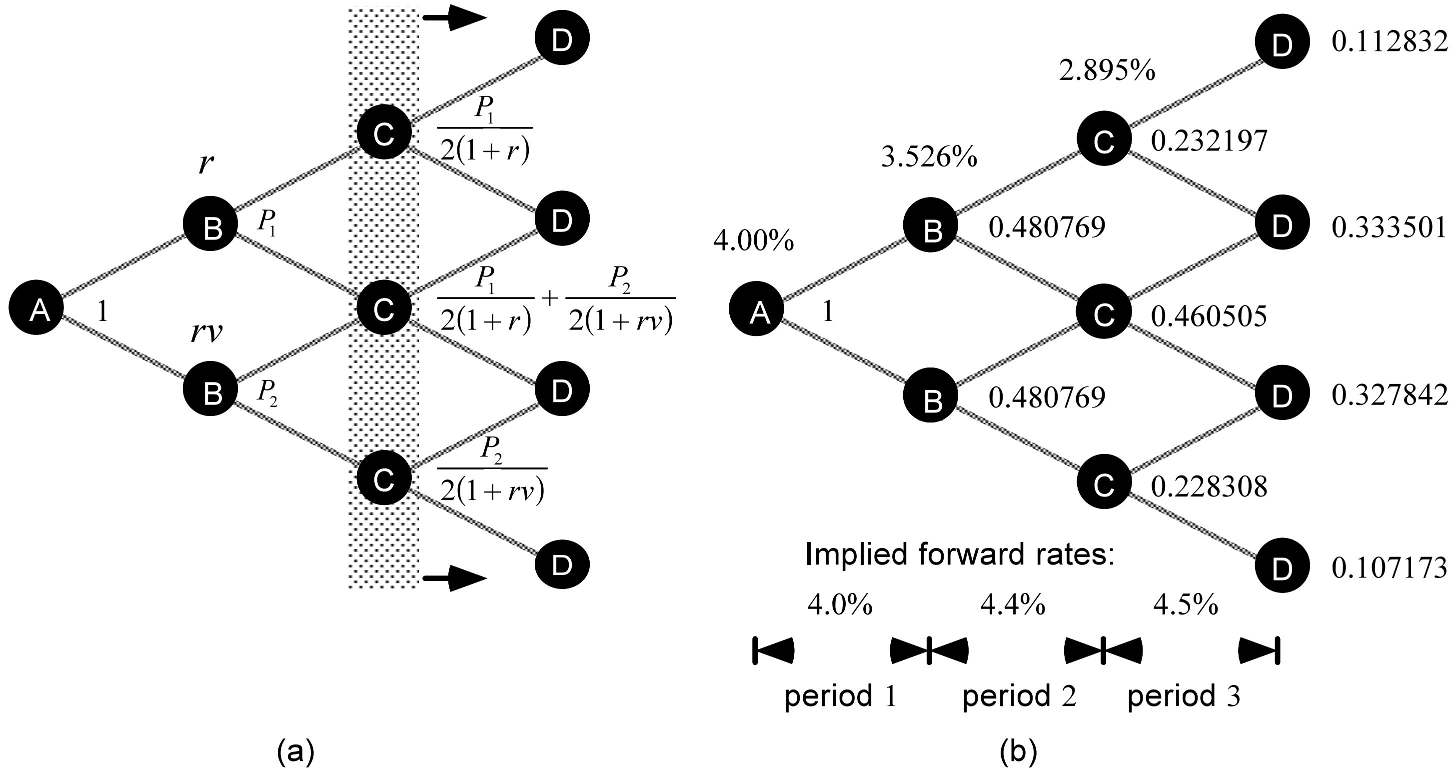
$$g(r) = \frac{1}{[\,1+S(j)\,]^j} \tag{137}$$

for r.

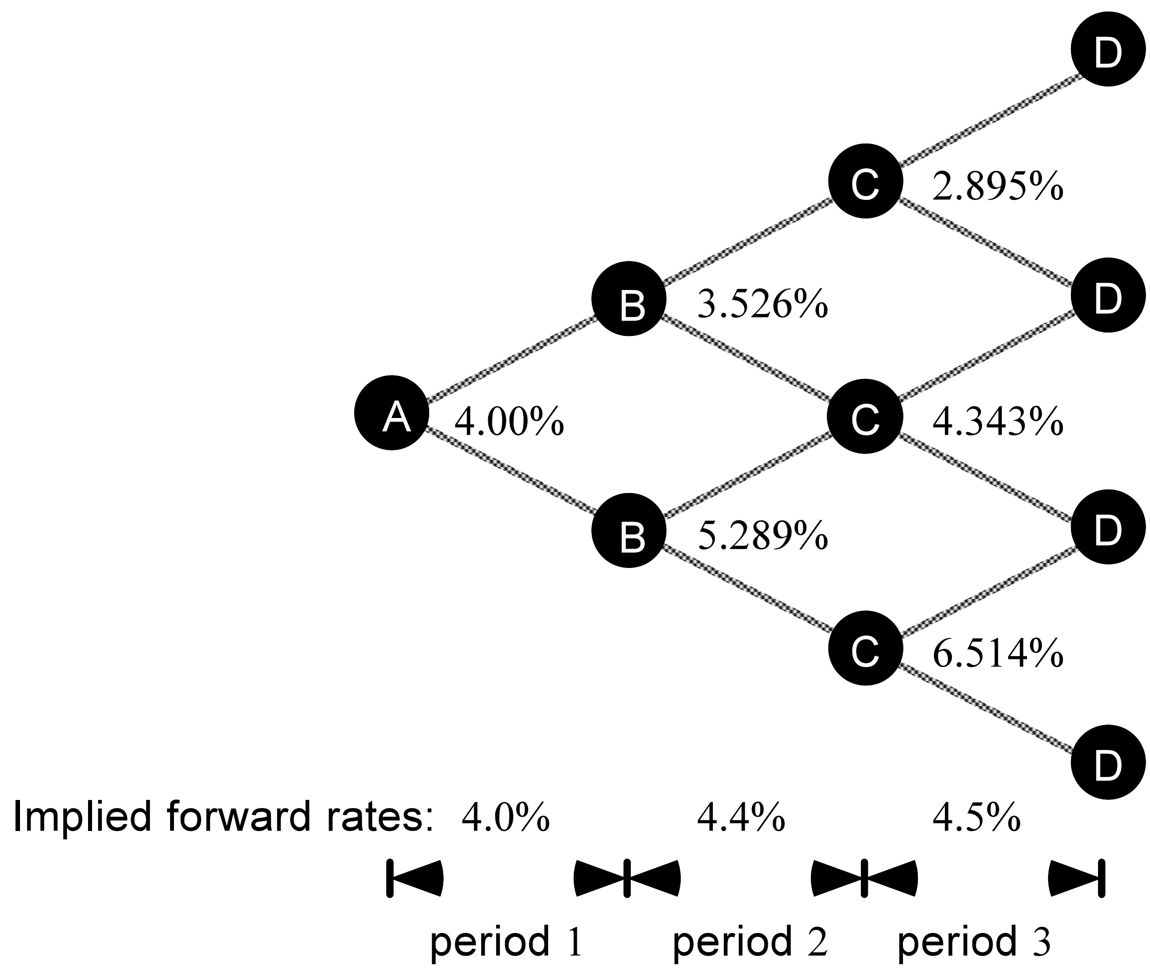

1
P_i
rv^{i-1}
1

Binomial Interest Rate Tree Calibration (continued)

- Given a decreasing market discount function, a unique positive solution for r is guaranteed.
- The state prices at time j can now be calculated (see panel (a) next page).
- We call a tree with these state prices a binomial state price tree (see panel (b) next page).
- The calibrated tree is depicted on p. 1043.



r
rv
A
1
B
P_1
P_2
C
D
$\frac{P_1}{2(1+r)}$
$\frac{P_1}{2(1+r)} + \frac{P_2}{2(1+rv)}$
$\frac{P_2}{2(1+rv)}$
(a)
4.00%
3.526%
2.895%
0.480769
0.480769
0.232197
0.460505
0.228308
0.112832
0.333501
0.327842
0.107173
Implied forward rates:
4.0%
4.4%
4.5%
period 1
period 2
period 3
(b)


D
C
2.895%
D
B
3.526%
A
4.00%
C
4.343%
D
B
5.289%
C
6.514%
D
Implied forward rates: 4.0%
4.4%
4.5%
period 1
period 2
period 3

Binomial Interest Rate Tree Calibration (concluded)

- The Newton-Raphson method can be used to solve for the r in Eq. (137) on p. 1039 as $g'(r)$ is easy to evaluate.
- The monotonicity and the convexity of $g(r)$ also facilitate root finding.
- The total running time is $O(n^2)$, as each root-finding routine consumes $O(j)$ time.
- With a good initial guess,[a] the Newton-Raphson method converges in only a few steps.[b]

---

[a]Such as $r_j = (\frac{2}{1+v_j})^{j-1} f_j$ on p. 1031.

[b]Lyuu (1999).

A Numerical Example

- One dollar at the end of the second period should have a present value of 0.92101 by 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 panel (b) on p. 1042 as 0.232197, 0.460505, and 0.228308.
- Their sum gives the correct market discount factor 0.92101.

A Numerical Example (concluded)

- 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 the calculation on p. 1032 using the new rates:

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

  which equals 0.88135, an exact match.

- The tree on p. 1043 prices without bias the benchmark securities.