3.25

Multi-period binomial model

$N+1$ times: $0, 1, 2 \ldots N$

Bank account:

r : (one-step) interest rate

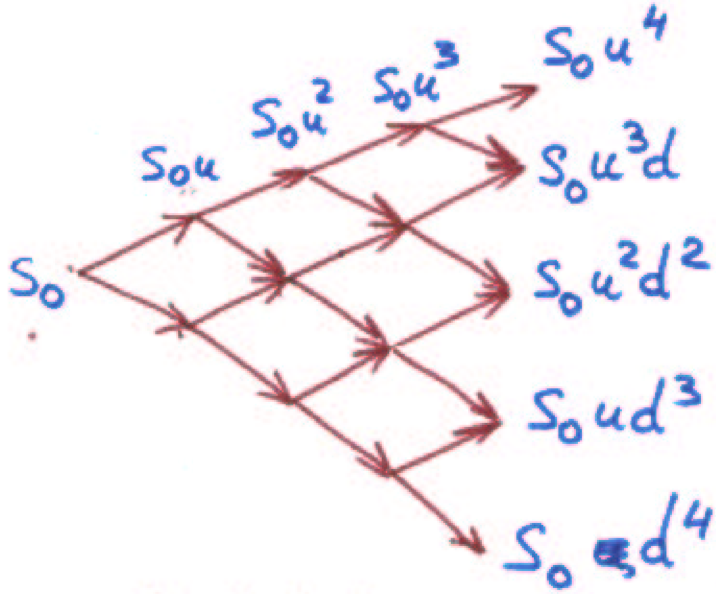
$$\begin{tikzcd} X_n \arrow[r, bend left] & X_n(1+r) \\ n & n+1 \end{tikzcd}$$

Stock :

$$\begin{tikzcd} & uS_n \\ S_n \arrow[ur, "H"] \arrow[dr, "T"'] & \\ & dS_n \\ n & n+1 \end{tikzcd}$$

3.26

$N = 4$



Probability space: $(\Omega, \mathbb{P})$

$|\Omega| = 2^N$

$\omega = (\omega_1 \dots \omega_N)$, $\omega_i \in \{H, T\}$

$\mathbb{P}[\omega] > 0$, $\omega \in \Omega$

3.27

Definition A sequence of random variables

$$X = (X_n)_{0 \leq n \leq N}$$

is called an adapted process if for any $0 \leq n \leq N$

$$X_n = X_n(\omega_1 \dots \omega_n)$$

(X_n depends on the evolution up to time n)

3.28

Problem Which of the following sequences is an adapted process?

1. $X_n = S_n\,,\quad 0 \le n \le N$

2. $X_n = S_N\,,\quad 0 \le n \le N$

3. $X_n = \max\limits_{0 \le k \le n} S_k\,,\quad 0 \le n \le N$

4. $X_n = S_{n+1} - S_n\,,\quad 0 \le n \le N$

Arbitrage

Lemma In the multi-period binomial model

$$(NA) \Longleftrightarrow d < 1+r < u$$

Proof

$\exists$ of arbitrage in the N-period model

$$\Updownarrow$$

$\exists$ of arbitrage for at least one branch.

$\blacksquare$

European options

3.30

An European option with maturity N is determined by its payoff:

$$V_N = V_N(\omega) = V_N(\omega_1 \ldots \omega_N)$$

Examples

Put : $V_N = \max(K - S_N, 0)$

Asian call : $V_N = \max\left(\frac{1}{N+1}\sum_{k=0}^{N} S_k - K, 0\right)$

Lookback put :

$$V_N = \max\left(K - \min_{k=\overline{0,N}} S_k, 0\right)$$

AFP of European options

Consider a European option with maturity N and payment

$$V_N = V_N(\omega), \quad \omega \in \Omega$$

AFP = Replication

Replication strategy:

$$\underset{(?)}{X_0} \longrightarrow \underset{(\text{known})}{X_N = V_N}$$

3.32

Better picture:

$$X_0 \to X_1 \to X_2 \to X_3 \cdots X_{N-1} \to \underbrace{X_N = V_N}_{\text{(known)}}$$

To compute a replication strategy we move BACKWARD.

boundary condition

$$\underbrace{X_n}_{\text{unknown}} \leftarrow \underbrace{X_{n+1} \cdots X_{N-1} \leftarrow X_N = V_N}_{\text{known}}$$

one-step induction:

X_n (unknown) $\qquad$ X_{n+1} (known)

We know (from previous computations)

$$X_{n+1} = X_{n+1}(\omega_1 \ldots \omega_n, \omega_{n+1})$$
$$\forall\ (\omega_1 \ldots \omega_n\ \omega_{n+1})$$

We need to compute

$$X_n = X_n(\omega_1 \ldots \omega_n)$$
$$\forall\ (\omega_1 \ldots \omega_n)$$

3.34

Denote

$\Delta_n = \Delta_n(\omega_1 \ldots \omega_n)$: # of stocks in a replication strategy at time n given $(\omega_1 \ldots \omega_n)$

Balance equation:

$$X_{n+1} = (X_n - \Delta_n S_n)(1+r) + \Delta_n S_{n+1}$$

Fix a trajectory $\omega_1 \ldots \omega_n$

3.35

We have

$$X_{n+1}(\omega_{n+1}=H) = (X_n - \Delta_n S_n)(1+r) + \Delta_n S_n u$$

$$X_{n+1}(\omega_{n+1}=T) = (X_n - \Delta_n S_n)(1+r) + \Delta_n S_n d$$

Same system as in one-period case!

$$\Delta_n(\omega_1 .. \omega_n) = \frac{X_{n+1}(\omega_{n+1}=H) - X_{n+1}(\omega_{n+1}=T)}{S_n(u-d)}$$

$$X_n(\omega_1 .. \omega_n) = \frac{1}{1+r}\left[\tilde{p}\, X_{n+1}(\omega_{n+1}=H) + \right.$$

$\tilde{q} \; X_{n+1} \; (\omega_{n+1} = T) \Big]$

8.36

where

$$\tilde{p} = \frac{1+r-d}{u-d}$$

$$\tilde{q} = 1-\tilde{p} = \frac{u-(1+r)}{u-d}$$

Theorem In multi-period binomial model the wealth process of replication strategy satisfies the algorithm of backward induction: 3.34

(a) Boundary condition:

$$X_N(\omega) = V_N(\omega), \quad \omega \in \Omega$$

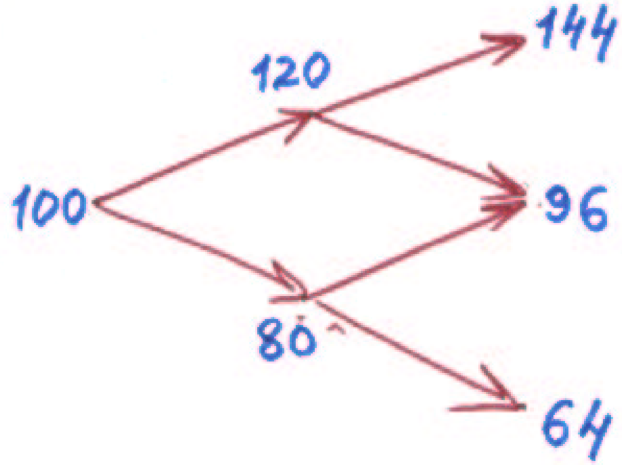
(b) One-step induction:

$$X_n(\omega_1 \dots \omega_n) = \frac{1}{1+r}\left[\tilde{p}\, X_{n+1}(\omega_1 \dots \omega_n H) + \tilde{q}\, X_{n+1}(\omega_1 \dots \omega_n T)\right]$$

$$\tilde{p} = \frac{1+r-d}{u-d} \qquad \tilde{q} = \frac{u-(1+r)}{u-d}$$

Problem

3.38



$N = 2$

$r = 10\%$

$u = 1.2$

$d = 0.8$

Compute arbitrage-free price for standard call $K = \$100$

Solution Compute 3.39

$\tilde{p}$ and $\tilde{q}$.

$$\tilde{p} = \frac{1+r-d}{u-d} = \frac{3}{4} \quad \tilde{q} = \frac{1}{4}$$

Baundary condition:

$$X_2(HH) = \max(S_2(HH)-K, 0)$$
$$= 44$$
$$X_2(HT) = 0$$
$$X_2(TH) = 0$$
$$X_2(TT) = 0$$

3.40

Backward induction

$$X_1 \longleftarrow X_2$$

$$X_1(H) = \frac{1}{1+r}\Big[X_2(HH)\tilde{p} + X_2(HT)\tilde{q}\Big] = 30$$

$$X_1(T) = \frac{1}{1+r}\Big[X_2(TH)\tilde{p} + X_2(TT)\tilde{q}\Big] = 0$$

$$X_0 \longleftarrow X_1$$

$$X_0 = \frac{1}{1+r}\Big[X_1(H)\tilde{p} + X_1(T)\tilde{q}\Big] = 20.4$$

Problem

$$
\begin{tikzcd}[row sep=small]
 & & 144 \\
 & 120 \arrow[ur] \arrow[dr] & \\
100 \arrow[ur] \arrow[dr] & & 96 \\
 & 80 \arrow[ur] \arrow[dr] & \\
 & & 64
\end{tikzcd}
\qquad r = 10\%
$$

Compute arbitrage-free price for Asian call:

$$V_2 = \max\left(\frac{1}{3}(S_0+S_1+S_2) - K, 0\right)$$

where $K = \$100$

Solution 342

$$\tilde{p} = \frac{1+r-d}{u-d} = \frac{3}{4} \quad \tilde{q} = \frac{1}{4}$$

Time 2 :

$$X_2(HH) = \max\left(\frac{100+120+144}{3} - 100, 0\right) = 21.33$$

$$X_2(HT) = 5.33$$

$$X_2(TH) = 0$$

$$X_2(TT) = 0$$

3.43

Time 1 :

$$X_1(H) = \frac{1}{1+r}\left[\tilde{p}\, X_2(HH) + \tilde{q}\, X_2(HT)\right]$$

$$= 15.45$$

$$X_1(T) = \frac{1}{1+r}\left[\tilde{p}\, X_2(TH) + \tilde{q}\, X_2(TT)\right]$$

$$= 0$$

Time 0 :

$$X_0 = \frac{1}{1+r}\left[X_1(H)\,\tilde{p} + X_1(T)\,\tilde{q}\right]$$

$$= 10.74$$

State processes 3.44

Question: Is the general algorithm of backward induction practical?

Answer: NO.

Indeed if we price an option with maturity 1 year and choose model, where

1 step = 1 working day

then

$N = 256$

3.45

In this case, to write boundary conditions we need to store

$$2^N = 2^{256} = \infty$$

values.

Idea: make algorithm of backward induction dependent on a type of a non-traded security.

Example Consider standard European option 3.46

$$V_N = f_N(S_N)$$

$f_N = f_N(x)$: deterministic function

$f_N(x) = \max(x-K, 0)$ call

$f_N(x) = I(x>K)$ digital

$f_N(x) = \max(x-K, K-x)$ straddle

3.44

claim:

$$X_n = f_n(S_n), \quad 0 \le n \le N,$$

$$f_n(x) = \frac{1}{1+r}\left\{\tilde{p}\, f_{n+1}(ux) + \tilde{q}\, f_{n+1}(dx)\right\},$$

$$\underbrace{x \in \{S_0 d^n, S_0 d^{n-1} u, \dots, S_0 d u^{n-1}, S_0 u^n\}}_{n+1 \text{ equation}}$$

$$n+1 \ll 2^n$$

Proof Backward induction: 3.48

$n=N$: by replication

$$X_N = V_N = f_N(S_N)$$

Assume that the representation

$$X_n = f_n(S_n)$$

is valid for $n \geq m+1$

Then $\forall\ (\omega_1 \ldots \omega_m)$

$$X_m(\omega_1 \ldots \omega_m) =$$

$$\frac{1}{1+r}\left[\tilde{p}\, X_{m+1}(\omega_1 \ldots \omega_m H) +\right.$$

$\boxed{3.49}$

$$\tilde{q}\, X_{m+1}(\omega_1 \ldots \omega_m T)\Big]$$

$$= \frac{1}{1+r}\Big[\tilde{p}\, f_{m+1}(S_m u) + \tilde{q}\, f_{m+1}(S_m d)\Big]$$

$$= f_m(S_m)$$

Hence

$$X_m(\omega_1 \ldots \omega_m) = f_m(S_m)$$

$\blacksquare$

$\boxed{3.50}$

$S = (S_n)_{0 \le n \le N}$ is an example of a state process

Definition An adapted process $Y = (Y_n)_{0 \le n \le N}$ is called a state process if $\forall$ European option

maturity : $m \le N$

payoff : $V_m = f_m(Y_m)$,

where

$f_m = f_m(x)$: deterministic function

we have $\forall\ 0 \leq n \leq m$ 3.51

X_n : capital of a replication strategy at n

$$X_n = f_n(Y_n)$$

for some deterministic function $f_n = f_n(x)$

Remark

$$\boxed{\text{\# of computations at time } n} = \boxed{\text{\# of different values of } Y_n}$$

3.52

General method:

given European option with payoff

$$V_N = V_N(\omega_1 \dots \omega_N)$$

find a state process $Y = (Y_n)_{0 \le n \le N}$ such that

$$V_N = f_N(Y_N)$$

for some deterministic $f_N = f_N(x)$.

Then (automatically!)

$$V_n = f_n(Y_n) \quad 0 \le n \le N$$

3.53

Remark. A state process is not defined uniquely.

Art of financial computations: choose state process to minimize the amount computations.

Convenient criterion

3.54

Proposition Consider an adapted process $X = (X_n)_{0 \le n \le N}$. Assume that for $\forall\, 1 \le n \le N$ $\exists$ deterministic functions $g_n = g_n(x)$ and $h_n = h_n(x)$ such that

$$X_n = \begin{cases} g_n(X_{n-1}) & \omega_n = H \\ h_n(X_{n-1}) & \omega_n = T \end{cases}$$

3.55

Then $X = (X_n)_{0 \le n \le N}$ is a state process.

Remark This condition is also necessary if $\tilde{p} \neq 0.5$

Intuitive description:

$\boxed{\text{knowledge of } X_n \text{ and } \omega_{n+1}} \Rightarrow \boxed{X_{n+1}}$

3.56

Proof Consider a European option with payoff

$$V_N = f_N(X_N)$$

Denote

V_n : capital of replication strategy at n

Claim:

$$V_n = f_n(X_n)$$

for some deteministic function $f_n = f_n(x)$.

Indeed, assume that this representation holds true for the time $n+1$: 3.54

$$V_{n+1} = f_{n+1}(X_{n+1})$$

From general algorithm of backward induction we deduce

$$V_n(\omega_1 \ldots \omega_n) = \frac{1}{1+r}\left[\tilde{p} * V_{n+1}(\omega_1 \ldots \omega_n H) + \tilde{q}\, V_{n+1}(\omega_1 \ldots \omega_n T)\right]$$

$$= \frac{1}{1+r}\left[\tilde{p}\, f_{n+1}(g_{n+1}(X_n)) + \right.$$

$$+\tilde{q}\, f_{n+1}\big(h_{n+1}(X_n)\big)\Big]$$

Hence,

$$V_n(\omega_1 \ldots \omega_n) = f_n(X_n), \text{ where}$$

$$f_n(x) = \frac{1}{1+r}\Big[\tilde{p}\, f_{n+1}\big(g_{n+1}(x)\big) + \tilde{q}\, f_{n+1}\big(h_{n+1}(x)\big)\Big]$$

∎

3.59

Problem Which of the following adapted processes is a state process?

1. $(S_n)_{0 \le n \le N}$ (stock)

2. $A = (A_n)_{0 \le n \le N}$, where
$$A_n = \frac{1}{n+1} \sum_{i=0}^{n} S_i$$

3. $(S_n, A_n)_{0 \le n \le N}$, where
$$A_n = \frac{1}{n+1} \sum_{i=0}^{n} S_i$$

3.60

4. $(M_n)_{0 \le n \le N}$, where

$$M_n = \max_{0 \le k \le n} S_k$$

5 $(M_n, S_n)_{0 \le n \le N}$

6. $(Z_n)_{0 \le n \le N}$, where

$$Z_n = I\left(\max_{0 \le k \le n} S_k \ge U\right)$$

~~lower~~ upper barrier

7. $(Z_n, S_n)_{0 \le n \le N}$, where 3.61

$$Z_n = I\left(\max_{0 \le k \le n} S_k \ge U\right)$$

2

Solution

3.62

1 YES

$$S_{n+1} = \begin{cases} S_n u & \omega_{n+1} = H \\ S_n d & \omega_{n+1} = T \end{cases}$$

(know S_n and $\omega_{n+1} \Rightarrow$
know S_{n+1})

2 NO

$$A_{n+1} = \frac{1}{n+1}\left(n A_n + S_{n+1}\right)$$

3.63

3. YES

$$(S_{n+1}, A_{n+1}) = \begin{cases} \left(S_n u, \frac{1}{n+1}(nA_n + S_n u)\right), & \omega_{n+1} = H \\ \left(S_n d, \frac{1}{n+1}(nA_n + S_n d)\right), & \omega_{n+1} = T \end{cases}$$

4. NO

$$M_{n+1} = \max(M_n, S_{n+1})$$

3.64

5. YES

$$(M_{n+1}, S_{n+1}) = \begin{cases} (\max(M_n, S_n u), S_n u), & \omega_{n+1} = H \\ (\max(M_n, S_n d), S_n d), & \omega_{n+1} = T \end{cases}$$

6. NO

$$Z_{n+1} = Z_n + (1 - Z_n) * I(S_{n+1} \geq \text{[illegible]})$$

7. YES

Solution 1 As 3.66

$$V_N = \max(S_N - K, 0)\, I(M_N \geq L)$$

where

$$M_n = \max_{0 \leq k \leq n} S_k, \quad 0 \leq n \leq N,$$

and

$(S_n, M_n)_{0 \leq n \leq N}$ is a state process,

we have

$$V_n = f_n(S_n, M_n), \quad 0 \leq n \leq N$$

capital of replicating strategy at n

where 3.67

$$f_N(s,m) = \max(s-K,0)\,I(m\geq L)$$

and

$$f_n(s,m) = \frac{1}{1+r}\Big[\tilde{p}\, f_{n+1}(us, \max(m, us)) + \tilde{q}\, f_{n+1}(ds, \max(m, ds))\Big], \quad 0 \leq n < N,$$

At time n we need to perform

$$\approx \frac{n^2}{2}$$

computations

Solution 2 As 3.68

$$V_N = \max(S_N - K, 0)\, Z_N$$

where

$$Z_n = I\left(\max_{0 \le k \le n} S_k \ge L\right)$$

and

$$(S_{n+1}, Z_{n+1}) = \begin{cases} (uS_n, & Z_n + (1-Z_n)\, I(uS_n \ge L), \text{ "H"} \\ (dS_n, & Z_n + (1-Z_n)\, I(dS_n \ge L, \text{ "T"} \end{cases}$$

we have that

3.69

$(S_n, Z_n)_{0 \le n \le N}$ is a state process and hence

$$V_n = f_n(S_n, Z_n), \quad 0 \le n \le N$$

We have

$$f_N(s,z) = \begin{cases} \max(s-K,0) & z=1 \\ 0 & z=0 \end{cases}$$

$$f_n(s,z) = \frac{1}{1+r}\Big[\tilde{p}\, f_{n+1}\big(us, I(us \ge L)\big) + \tilde{q}\, f_{n+1}\big(ds, I(ds \ge L)\big)\Big]$$

$$f_n(s,1) = \frac{1}{1+r}\left[\tilde{p} f_{n+1}(us,1) + \tilde{q} f_{n+1}(ds,1)\right]$$ 3.40

At time n we need to perform exactly
$$2(n+1)$$
computations.

Problem In $N=3$ 3.71

period binomial model with

$$S_0 = 4, \quad u = 2, \quad d = \frac{1}{2}$$

$$r = 0.25$$

compute the price of Asian call

$$V_3 = \max\left(\frac{1}{4} Y_3 - K, 0\right)$$

$$Y_n = \sum_{k=0}^{n} S_k \quad , \quad K = 4$$

Solution 3.42

(S_n, Y_n), $0 \le n \le N$,

is a state process,

because

$$(S_{n+1}, Y_{n+1}) = \begin{cases} (uS_n, Y_n + uS_n), & \omega_{n+1} = H \\ (dS_n, Y_n + dS_n), & \omega_{n+1} = T \end{cases}$$

Hence

$$V_n = f_n(S_n, Y_n), \quad 0 \le n \le N$$

$$f_N(s,y) = \max\left(\frac{1}{N+1}\,y - K, 0\right)$$

$\boxed{3.43}$

$$f_n(s,y) = \frac{1}{1+r}\left[\tilde{p}\, f_{n+1}(us, y+us) + \tilde{q}\, f_{n+1}(ds, y+ds)\right]$$

$$\tilde{p} = \frac{1+r-d}{u-d} = \frac{1}{2}$$

$$\tilde{q} = \frac{1}{2}$$

$$f_n(s,y) = \frac{2}{5}\Big(f_{n+1}(us, y+us) + f_{n+1}(ds, y+ds)\Big)$$

Step 1: (Move forward) 3.74

Compute the possible values for (S, Y).

Time 0: $\{(4,4)\}$

Time 1: $\{(8,12), (2,6)\}$

Time 2: $\{(16,28), (4,16), (4,10), (1,7)\}$

Time 3: $\{(32,60), (8,36), (8,24), (2,18), (8,18), (2,12), (2,9), (\frac{1}{2}, 7\frac{1}{2})\}$

Step 2 : (Move backward) 3 45

Backward induction:

$$\max\left(\frac{60}{4}-4, 0\right)$$

Time 3:

$$f_3(32, 60) = \left(\frac{60}{4} - 4\right)^+ = 11$$

$$f_3(8, 36) = \left(\frac{36}{4} - 4\right)^+ = 5$$

$$f_3(8, 24) = \left(\frac{24}{4} - 4\right)^+ = 2$$

$$f_3(8, 18) = \left(\frac{18}{4} - 4\right)^+ = \frac{1}{2}$$

$$f_3(2, 18) = \frac{1}{2}$$

$$f_3(2, 12) = 0$$

$$f_3(2, 9) = 0$$

$$f_3\left(\frac{1}{2}, 7\frac{1}{2}\right) = 0$$

376

Time 2:

$$f_2(16,28) = \frac{2}{5}\Big(f_3(32,60) + f_3(8,36)\Big) = 6.4$$

$$f_2(4,16) = \frac{2}{5}\Big(f_3(8,24) + f_3(2,18)\Big) = 1$$

$$f_2(4,10) = 0.2$$

$$f_2(1,7) = 0$$

Time 1:

$$f_1(8,12) = \frac{2}{5}\Big(f_2(16,28) + f_2(4,16)\Big) = 2.96$$

$$f_1(2,6) = \frac{2}{5}\Big(f_2(4,10) + f_2(1,7)\Big) = 0.08$$

3.77

Time 0:

$$V_0 = f_0(4,4) = \frac{2}{5}\Big(f_1(8,12) + f_1(2,6)\Big) = 1.216$$

Summary on state processes. 3.78

A "naive" algorithm of backward induction, where

$$\boxed{\text{\# of comutations at time } n} = \boxed{\text{\# of elementary events at time } n \; (2^n)}$$

is unpractical.

Idea: adapt this algorithm to the non-traded security we want to price.

B Method: represent the terminal payoff as

$$\underset{\text{payoff}}{V_N} = \underset{\substack{\text{deterministic}\\\text{function}}}{f_N}\,(\underset{\substack{\text{state}\\\text{process}}}{Y_N})$$

$$\boxed{\text{\# of computations at time } n} = \boxed{\text{\# of different values for } Y_n}$$

Goal: choose the state process with smallest set of values.