Options & Derivatives

The One Period Model

- time t: $t = 0, 1$
- (deterministic) bond B_t: $B_0 = 1,\ B_1 = 1 + R$
- (stochastic) stock S_t: $S_0 = s > 0,\ S_1 = \begin{cases} s \cdot u & \text{with prob. } p_u \\ s \cdot d & \text{with prob. } p_d \end{cases} \equiv s\,Z$:

  $u > d$, $p_u + p_d = 1$.
- The value V_t^h of the portfolio $h = (x, y)$, $x, y \in \mathbb{R}$ at time t:
  $V_t^h = x\,B_t + y\,S_t$ — $V_0^h = x + y\,s,\ V_1^h = x(1+R) + y\,s\,Z$
- Arbitrage portfolio h: $V_0^h = 0$, $V_1^h > 0$ with prob. 1.

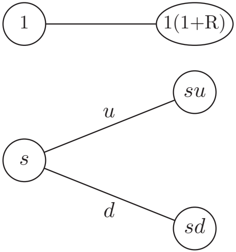


Figure: Asset Dynamics of One Period Model.

Portfolios and Arbitrage I

Theorem

The one period model is arbitrage free $\iff$ $u \geqslant 1 + R \geqslant d$.

Proof

$(\Longrightarrow)$

- Suppose $u \geqslant 1 + R \geqslant d$ does not hold, then $1 + R > u$ or $d > 1 + R$.
- If $1 + R > u$, then $s(1 + R) > s\,u$ and a priori $s(1 + R) > s\,d$.
- Consider $h = (s, -1)$, then $V_0^h = s \cdot 1 + (-1) \cdot s = 0$, $V_1^h = s(1 + R) - s \cdot Z > 0$, an arbitrage.
- If $d > 1 + R$, then $s\,d > s(1 + R)$ and a priori $s\,u > s(1 + R)$.
- Consider $h = (-s, 1)$, then $V_0^h = (-s) \cdot 1 + 1 \cdot s = 0$, $V_1^h = -s(1 + R) + s \cdot Z > 0$, an arbitrage.

Portfolios and Arbitrage II

Theorem

The one period model is arbitrage free $\iff$ $u \geqslant 1 + R \geqslant d$.

Proof

$(\Longleftarrow)$

- Arbitrage $h = (x, y)$: $V_0^h = 0$.
- $x + s \cdot y = 0 \implies x = -s \cdot y$.
- $V_1^h = \begin{cases} y\, s(u - (1+R)), & Z = u \\ y\, s(d - (1+R)), & Z = d \end{cases}$
- If $y > 0$: from $V_1^h > 0 \implies u > 1 + R$ and $d > 1 + R$; a contradiction.
- If $y < 0$: from $V_1^h > 0 \implies u < 1 + R$ and $d < 1 + R$; a contradiction.

Risk-Neutral / Martingale Measure and Probabilities

- Observation: $u \geqslant 1+R \geqslant d \implies 1+R$ is a convex combination of u and d
- $\exists\, q_u, q_d \geqslant 0,\ q_u + q_d = 1$ s.t. $1+R = q_u \cdot u + q_d \cdot d$
- Define a new probability measure Q and the associated expectation E^Q s.t.

$$Q(Z=u) = q_u, \quad Q(Z=d) = q_d$$
$$\frac{1}{1+R}\,\mathsf{E}^Q\, S_1 = \frac{1}{1+R}(q_u \cdot s\,u + q_d \cdot s\,d) = \frac{1}{1+R} \cdot s(1+R) = s$$

Definition

- **Risk-Neutral / Martingale Measure**: A measure Q satisfies $S_0 = \dfrac{1}{1+R}\,\mathsf{E}^Q\, S_1$.
- **Martingale Probabilities**: $q_u = \dfrac{(1+R)-d}{u-d},\ q_d = \dfrac{u-(1+R)}{u-d}$

Contingent Claims I

Definition

- A **contingent claim** X is of the form $X = \Phi(Z)$
- Stochastic Z with **contract function** $\Phi(\cdot)$
- **Price** of X at time t: $\Pi(t; X)$

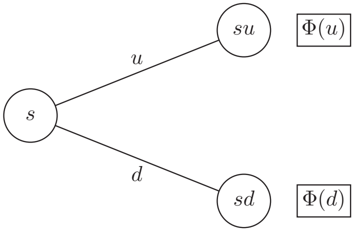


Figure: The Contingent Claim.

Contingent Claims II

Example (European Call Option with Strike K)

Assume $s\,u > K > s\,d$. At $t = 1$,

- Exercise the option if $S_1 > K$.
  - Pay K to get the stock and sell it at $s\,u$, thus making net profit $s\,u - K$.
- Do nothing if $S_1 < K$.

$$X = \begin{cases} s\,u - K, & Z = u \\ 0, & Z = d \end{cases}, \qquad \begin{cases} \Phi(u) = s\,u - K \\ \Phi(d) = 0 \end{cases}$$

Definition

- A contingent claim X is said to be **reachable** if there exists a portfolio h such that $V_1^h = X$ with probability 1; this portfolio h is called a **hedging** or **replicating** portfolio.
- If all claims can be replicated we say the market is **complete**.

Contingent Claims III

Theorem (Pricing Principle)

If a claim X is reachable with replicating portfolio h, then the "reasonable" price of X is given by $\Pi(t; X) = V_t^h,\ t = 0, 1.$

Theorem

An arbitrage free one period model is complete.

Proof

Fixed any $\Phi(\cdot)$, show that $\exists\, h = (x, y)$ s.t.

$$V_1^h = \begin{cases} \Phi(u) & Z = u, \\ \Phi(d) & Z = d. \end{cases} \implies x(1+R) + y\, s\, u = \Phi(u),\ x(1+R) + y\, s\, d = \Phi(d).$$

Solve for x, y: $x = \dfrac{1}{1+R} \dfrac{u\Phi(d) - d\,\Phi(u)}{u-d}, \quad y = \dfrac{1}{s} \dfrac{\Phi(u) - \Phi(d)}{u-d}.$

Risk Neutral Valuation

- From Pricing Principle ($\Pi(t; X) = V_t^h$, $t = 0, 1$)

$$\begin{aligned}
\Pi(0; X) = V_0^h &= x + s\, y \\
&= \frac{1}{1+R} \cdot \frac{u\Phi(d) - d\,\Phi(u)}{u-d} + s \cdot \frac{1}{s} \cdot \frac{\Phi(u) - \Phi(d)}{u-d} \\
&= \frac{1}{1+R} \left\{ \frac{(1+R) - d}{u-d}\, \Phi(u) + \frac{u - (1+R)}{u-d}\, \Phi(d) \right\} \\
&= \frac{1}{1+R} \{ q_u\, \Phi(u) + q_d\, \Phi(d) \} \equiv \frac{1}{1+R}\, \mathsf{E}^Q\, X
\end{aligned}$$

Theorem (The Risk Neutral Valuation Principle)

If the one period binomial model is arbitrage-free, then the price of X is $\Pi(0; X) = \dfrac{1}{1+R}\, \mathsf{E}^Q\, X$.

The Multiperiod Model

- time t: $t = 0, 1, 2, \dots, T$
- (deterministic) bond B_t with $B_0 = 1,\ B_{n+1} = (1 + R)B_n$
- (stochastic) stock S_t with $S_0 = s > 0,\ S_{n+1} = Z_n\, S_n$ where $Z_0, Z_1, Z_2, \dots, Z_{T-1}$ are iid with $\mathsf{P}(Z_n = u) = p_u,\ \mathsf{P}(Z_n = d) = p_d$

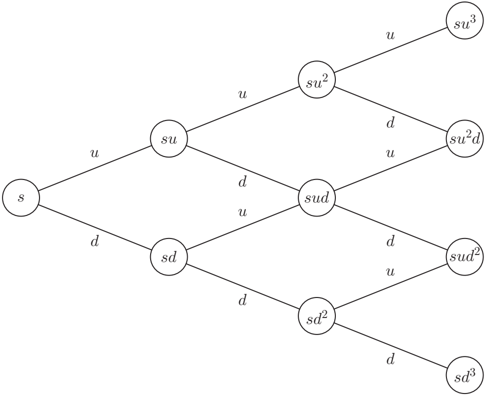


Figure: Asset Dynamics of Multiperiod Model: “Recombining” Tree.

Portfolios and Arbitrage

Definition

The portfolio $h_t \equiv (x_t, y_t)$; The value $V_t^{h_t}$ of portfolio h_t at time t is $V_t^{h_t} = x_t\, B_t + y_t\, S_t$.

- Hereafter we write V_t^h instead of the cumbersome $V_t^{h_t}$.
- x_t is the amount which we invest in the bank at time $t-1$ and keep until t.

Definition

Self-financing portfolio $h_t = (x_t, y_t)$:
$x_t\,(1+R) + y_t\, S_t = x_{t+1} + y_{t+1}\, S_t, \quad \forall t = 0, 1, \dots, T-1.$

Contingent Claims

Definition

- Arbitrage: there exists a self-financing portfolio h_t with $V_0^h = 0$, $\mathsf{P}(V_T^h \geqslant 0) = 1$, $\mathsf{P}(V_T^h > 0) > 0$.
- A contingent claim X is said to be **reachable** if there exists a self-financing portfolio h such that $V_T^h = X$ with probability 1; this portfolio h is called a **hedging** or **replicating** portfolio.
- If all claims can be replicated we say the market is **complete**.

Theorem (Pricing Principle)

If a claim X is reachable with replicating (and self-financing) portfolio h, then the "reasonable" price process of X is given by $\Pi(t; X) = V_t^h,\ t = 0, 1, 2, \ldots T$.

Theorem

An arbitrage-free multiperiod model is complete.

Theorem (Binomial Algorithms)

- Given a contingent claim $X = \Phi(S_T)$; let $V_t(k)$ denotes the value of the replicating portfolio at node (t, k), then $V_t(k)$ is computed recursively by

$$V_T(k) = \Phi(s\, u^k\, d^{T-k})$$
$$V_t(k) = \frac{1}{1+R}\left\{q_u\, V_{t+1}(k+1) + q_d\, V_{t+1}(k)\right\}$$

- The martingale probabilities q_u, q_d are $q_u = \dfrac{(1+R)-d}{u-d}$, $q_d = \dfrac{u-(1+R)}{u-d}$
- The replicating portfolio $h_t = (x_t, y_t)$ is

$$x_t(k) = \frac{1}{1+R}\,\frac{u\, V_t(k) - d\, V_t(k+1)}{u-d}, \quad y_t(k) = \frac{1}{S_{t-1}}\,\frac{V_t(k+1) - V_t(k)}{u-d}$$

- The arbitrage-free price of a contingent claim X at $t = 0$ is

$$\Pi(0; X) = \frac{1}{(1+R)^T}\,\mathsf{E}^Q X = \frac{1}{(1+R)^T}\cdot\sum_{k=0}^{T}\binom{T}{k} q_u^k\, q_d^{T-k}\,\Phi(s\, u^k\, d^{T-k})$$

Example

Given $T = 3, S_0 = 80, K = 80, u = 1.5, d = 0.5, p_u = 0.6, p_d = 0.4, R = 0$, compute the European call option price and the replicating portfolio of each node.

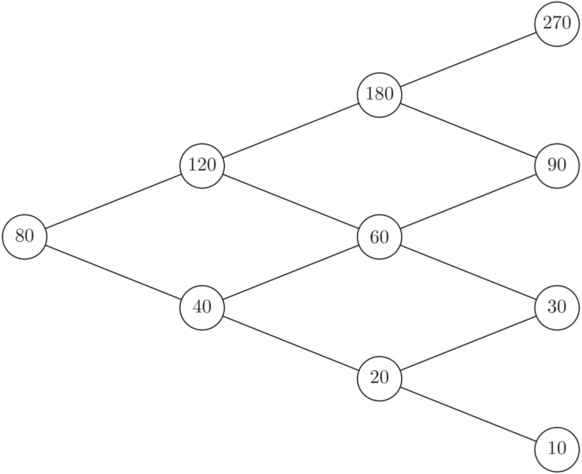


Figure: Asset Dynamics of the Example.

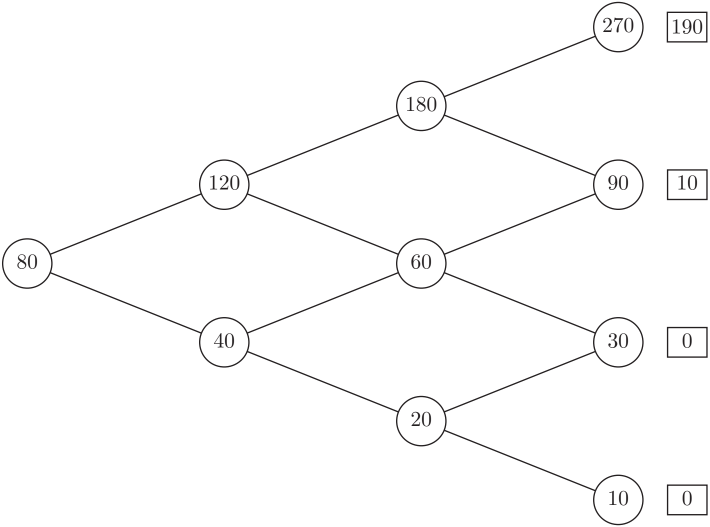


Figure: Payoff at the End of Terms.

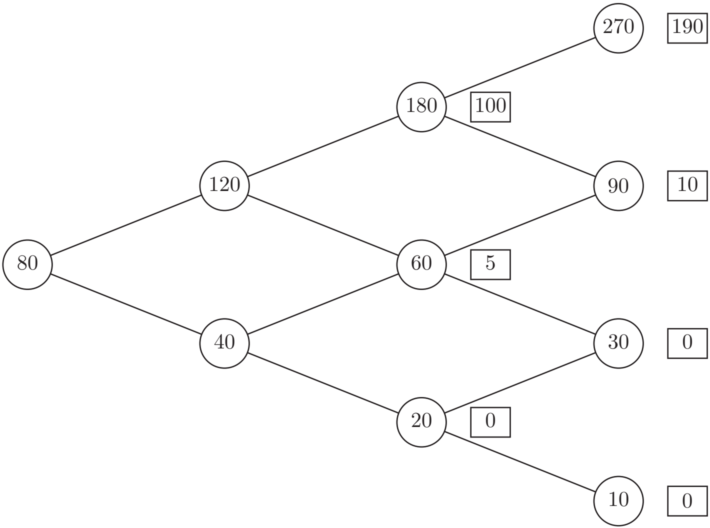


Figure: Iterated Computation of $\Pi(t;X):\ \Pi(t-1;X) \equiv \dfrac{1}{1+R}\,\mathsf{E}^{Q}\{\Pi(t;X)\}$.

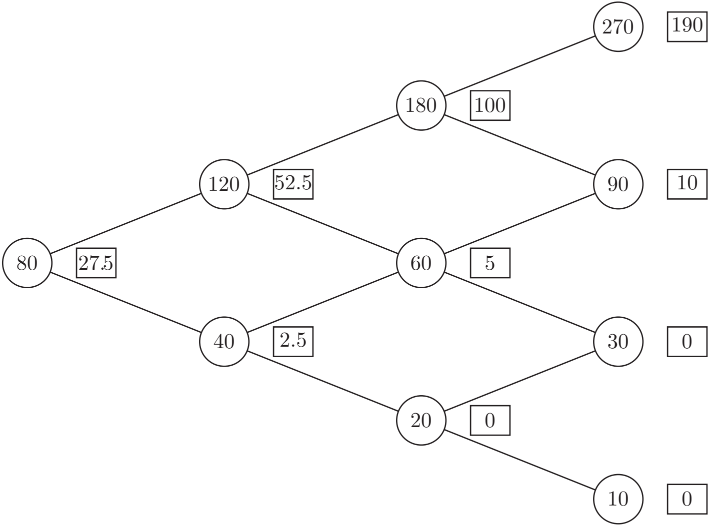


Figure: The Completed $\Pi(t; X)$.

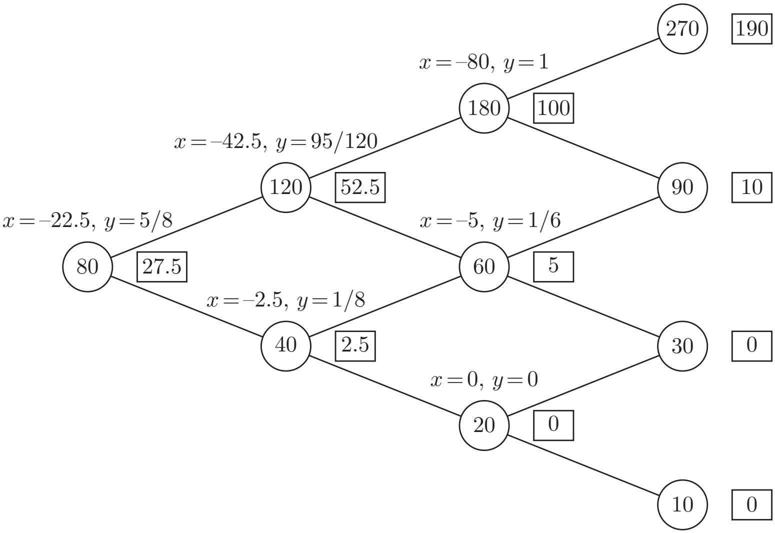


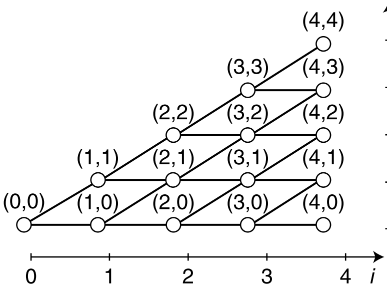
Figure: Replicating $h_t = (x_t, y_t)$: $x_t(k) = \frac{1}{1+R} \frac{u\,V_t(k) - d\,V_t(k+1)}{u-d}$, $y_t(k) = \frac{1}{S_{t-1}} \frac{V_t(k+1) - V_t(k)}{u-d}$

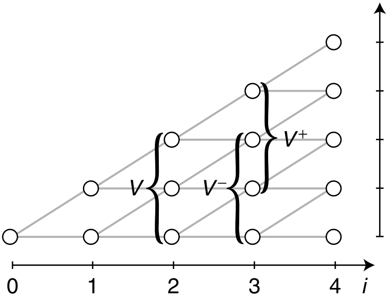
Algorithmic Considerations

$$\Pi(0; X) = \frac{1}{(1+R)^T} \cdot \sum_{k=0}^{T} \binom{T}{k} q_u^k \, q_d^{T-k} \Phi(s\, u^k\, d^{T-k})$$

For big T the formula can't be directly used because of the binomial coefficient

$$V_T(k) = \Phi(s\, u^k\, d^{T-k}), \quad V_t(k) = \frac{1}{1+R}\left\{q_u\, V_{t+1}(k+1) + q_d\, V_{t+1}(k)\right\}$$





Python Code Illustration: Common Parts

```python
import numpy as np

S0 = 80; r = 0; K = 80; u = 1.5; d = 0.5;
q = (1 - d) / (u - d); M = 3;
df = 1     # discount factor per time interval
# exhibit stock paths
S = np.zeros((M + 1, M + 1), dtype=np.float)
S[0, 0] = S0
for j in range(1, M + 1, 1):
    for i in range(j + 1):
        S[i, j] = S[0, 0] * (u ** (j - i)) * (d ** i)
```

Python Codes: Traditional Loops

```
iv = np.zeros((M + 1, M + 1), dtype=np.float); z = 0  # inner values
for j in range(0, M + 1, 1):
    for i in range(z + 1):
        iv[i, j] = round(max(S[i, j] - K, 0), 8)
    z += 1

pv = np.zeros((M + 1, M + 1), dtype=np.float)         # present values
pv[:, M] = iv[:, M]
z = M + 1
for j in range(M - 1, -1, -1):
    z -= 1
    for i in range(z):
        pv[i, j] = (q * pv[i, j + 1] + (1 - q) * pv[i + 1, j + 1]) * df
```

Python Codes: Vectorized Loops

```
import numpy as np
from params import *
import time

mu = np.arange(M + 1)
mu = np.resize(mu, (M + 1, M + 1))
md = np.transpose(mu)
mu = u ** (mu - md)
md = d ** md
S = S0 * mu * md

start_time = time.time()

# present value array initialized with inner values
pv = np.maximum(S - K, 0)
z = 0
for i in range(M - 1, -1, -1): # backwards induction
    pv[0:M-z, i] = (q * pv[0:M-z, i+1] + (1 - q) * pv[1:M-z+1, i+1]) * df
    z += 1

print(pv)
print('Value of European call option is %8.3f' % pv[0, 0])
print('vector elapsed: %f seconds.' % (time.time() - start_time,))
```