MATH97110: Numerical Methods for Finance

Topic 2: Path-dependent Options Pricing with Lattice Methods

Imperial College London

2021-2022

Overview

1. Introduction to path-dependent options pricing
2. Forward shooting grid method
3. Two specific examples:
    - Lookback option
    - Asian option

Path-dependent options

- In the last lecture, we focused on pricing an option under binomial tree when the payoff depends on the stock price at maturity only, i.e. the payoff function is in form of $g(S_N)$

Path-dependent options

- In the last lecture, we focused on pricing an option under binomial tree when the payoff depends on the stock price at maturity only, i.e. the payoff function is in form of $g(S_N)$
- In contrast, a path-dependent option has payoff function depending on the stock price at multiple time points in form of $g(S_0, S_1, ..., S_N)$

Path-dependent options

- In the last lecture, we focused on pricing an option under binomial tree when the payoff depends on the stock price at maturity only, i.e. the payoff function is in form of $g(S_N)$
- In contrast, a path-dependent option has payoff function depending on the stock price at multiple time points in form of $g(S_0, S_1, ..., S_N)$
- Examples:

Floating strike lookback put option: $\left(\max_{i=0,\dots,N} S_i - S_N\right)$

Fixed strike Asian call option: $\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^+$

Path-dependent options

- In the last lecture, we focused on pricing an option under binomial tree when the payoff depends on the stock price at maturity only, i.e. the payoff function is in form of $g(S_N)$
- In contrast, a path-dependent option has payoff function depending on the stock price at multiple time points in form of $g(S_0, S_1, ..., S_N)$
- Examples:

Floating strike lookback put option: $\left(\max_{i=0,\dots,N} S_i - S_N\right)$

Fixed strike Asian call option: $\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^+$

- Suppose the payoff is paid to the option holder at the terminal time N, then the time-zero fair option price is

$$e^{-rN\triangle t}\mathbb{E}_{\mathbb{Q}}[g(S_0, S_1, ..., S_N)] \tag{2.1}$$

where r is the (annualised) interest rate and $\triangle t$ is the calender time (in year) corresponding to each period of the tree

A naive approach to price a path-dependent option

- A direct way to compute the expectation in (2.1) is to loop through all possible realisation of paths
- Example: Pricing a fixed strike Asian call option in a 3-period binomial tree with $S_0 = 100$, $K = 100$, $u = 2$, $d = 0.5$ and $q = 0.6$, and interest rate $r = 0$

Path scenario	S_1	S_2	S_3	probability	payoff $= \left(\frac{1}{4}\sum_{i=0}^{3} S_i - K\right)^+$
up up up	200	400	800	$q^3 = 0.216$	275
up up down	200	400	200	$q^2(1-q) = 0.144$	125
up down up	200	100	200	$q^2(1-q) = 0.144$	50
up down down	200	100	50	$q(1-q)^2 = 0.096$	12.5
down up up	50	100	200	$q^2(1-q) = 0.144$	12.5
down up down	50	100	50	$q(1-q)^2 = 0.144$	0
down down up	50	25	50	$q(1-q)^2 = 0.096$	0
down down down	50	25	12.5	$(1-q)^3 = 0.064$	0

and the fair option price is $e^{-rN\triangle t}\left(\sum_i \text{probability}_i \times \text{payoff}_i\right) = 87.6$

A naive approach to price a path-dependent option

- A direct way to compute the expectation in (2.1) is to loop through all possible realisation of paths
- Example: Pricing a fixed strike Asian call option in a 3-period binomial tree with $S_0 = 100$, $K = 100$, $u = 2$, $d = 0.5$ and $q = 0.6$, and interest rate $r = 0$

Path scenario	S_1	S_2	S_3	probability	payoff $= \left(\frac{1}{4}\sum_{i=0}^{3} S_i - K\right)^+$
up up up	200	400	800	$q^3 = 0.216$	275
up up down	200	400	200	$q^2(1-q) = 0.144$	125
up down up	200	100	200	$q^2(1-q) = 0.144$	50
up down down	200	100	50	$q(1-q)^2 = 0.096$	12.5
down up up	50	100	200	$q^2(1-q) = 0.144$	12.5
down up down	50	100	50	$q(1-q)^2 = 0.144$	0
down down up	50	25	50	$q(1-q)^2 = 0.096$	0
down down down	50	25	12.5	$(1-q)^3 = 0.064$	0

and the fair option price is $e^{-rN\triangle t}\left(\sum_i \text{probability}_i \times \text{payoff}_i\right) = 87.6$

- In an N-period binomial tree, there are 2^N possible realisations of the stock price path $(S_0, S_1, ..., S_N)$ and looping through all possible scenarios is not feasible for large N

Pricing path-dependent option: introducing an auxiliary process

- For many types of path-dependent options, the payoff function $g(S_0, S_1, ..., S_N)$ can be simplified by introducing a new auxiliary state process $F = (F_n)_{n=0,\dots,N}$, and then the payoff function can be rewritten as $g(S_N, F_N)$

Pricing path-dependent option: introducing an auxiliary process

- For many types of path-dependent options, the payoff function $g(S_0, S_1, ..., S_N)$ can be simplified by introducing a new auxiliary state process $F = (F_n)_{n=0,\dots,N}$, and then the payoff function can be rewritten as $g(S_N, F_N)$
- Example: Floating strike lookback put option $\left(\max_{i=0,1,\dots,N} S_i - S_N\right)$
  - Define $M_n := \max_{i=0,1,\dots,n} S_i$ which represents the running maximum of the stock price process between time zero and time n, then the payoff function becomes

$$g(S_N, M_N) = M_N - S_N$$

Pricing path-dependent option: introducing an auxiliary process

- For many types of path-dependent options, the payoff function $g(S_0, S_1, ..., S_N)$ can be simplified by introducing a new auxiliary state process $F = (F_n)_{n=0,\dots,N}$, and then the payoff function can be rewritten as $g(S_N, F_N)$
- Example: Floating strike lookback put option $\left(\max\limits_{i=0,1,\dots,N} S_i - S_N\right)$
  - Define $M_n := \max_{i=0,1,\dots,n} S_i$ which represents the running maximum of the stock price process between time zero and time n, then the payoff function becomes

$$g(S_N, M_N) = M_N - S_N$$

- Example: Fixed strike Asian call option $\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^+$
  - Define $A_n := \frac{1}{n+1}\sum_{i=0}^{n} S_i$ which represents the running average of the stock price process between time zero and time n, then the payoff function becomes

$$g(A_N) = (A_N - K)^+$$

Pricing path-dependent option: introducing an auxiliary process

- For many types of path-dependent options, the payoff function $g(S_0, S_1, ..., S_N)$ can be simplified by introducing a new auxiliary state process $F = (F_n)_{n=0,\ldots,N}$, and then the payoff function can be rewritten as $g(S_N, F_N)$
- Example: Floating strike lookback put option $\left(\max\limits_{i=0,1,\ldots,N} S_i - S_N\right)$
  - Define $M_n := \max_{i=0,1,\ldots,n} S_i$ which represents the running maximum of the stock price process between time zero and time n, then the payoff function becomes

$$g(S_N, M_N) = M_N - S_N$$

- Example: Fixed strike Asian call option $\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^+$
  - Define $A_n := \frac{1}{n+1}\sum_{i=0}^{n} S_i$ which represents the running average of the stock price process between time zero and time n, then the payoff function becomes

$$g(A_N) = (A_N - K)^+$$

- Now the path-dependent nature of the payoff disappears, but this comes at the cost of analysing an additional process

The pricing algorithm for path-dependent options

- If the payoff can be rewritten as $g(S_N, F_N)$ for some suitably chosen auxiliary process F, then the fair price of the path-dependent option at time n becomes

$$V^n := e^{-r(N-n)\triangle t}\mathbb{E}_{\mathbb{Q}}\left[g(S_N, F_N)\middle|\mathcal{F}_n\right]$$

The pricing algorithm for path-dependent options

- If the payoff can be rewritten as $g(S_N, F_N)$ for some suitably chosen auxiliary process F, then the fair price of the path-dependent option at time n becomes

$$V^n := e^{-r(N-n)\triangle t}\mathbb{E}_{\mathbb{Q}}\left[g(S_N, F_N)\middle|\mathcal{F}_n\right]$$

- The backward induction algorithm for computing option price remains largely the same as in Prop 1.1 of Topic 1:

$$V^n = \begin{cases} g(S_N, F_N), & n = N; \\ e^{-r\triangle t}\mathbb{E}_{\mathbb{Q}}\left[V^{n+1}\middle|\mathcal{F}_n\right], & n = 0, 1, ..., N-1 \end{cases}$$

The pricing algorithm for path-dependent options

- If the payoff can be rewritten as $g(S_N, F_N)$ for some suitably chosen auxiliary process F, then the fair price of the path-dependent option at time n becomes

$$V^n := e^{-r(N-n)\triangle t}\mathbb{E}_{\mathbb{Q}}\left[g(S_N, F_N)\middle|\mathcal{F}_n\right]$$

- The backward induction algorithm for computing option price remains largely the same as in Prop 1.1 of Topic 1:

$$V^n = \begin{cases} g(S_N, F_N), & n = N; \\ e^{-r\triangle t}\mathbb{E}_{\mathbb{Q}}\left[V^{n+1}\middle|\mathcal{F}_n\right], & n = 0, 1, ..., N-1 \end{cases}$$

- The main challenge is to take into account the random transition of F from time n to $n+1$ as well when calculating $\mathbb{E}_{\mathbb{Q}}\left[V^{n+1}\middle|\mathcal{F}_n\right]$

Pricing path-dependent options in a binomial tree

- Recall that in a binomial tree:
  - We set
    $$s_k^n = S_0 u^{n-k} d^k, \qquad \text{for } n = 0, 1, ..., N \text{ and } k = 0, 1, ..., n$$
    as the possible values of stock price at each time point
  - The transition dynamics of S is known precisely: when $S_n = s_k^n$ moves to $s_{k_{new}}^{n+1}$, then $k_{new} = k$ or $k_{new} = k + 1$
  - For a simple non-path-dependent option, if we denote V_k^n as the time-n option price when the current stock price is s_k^n then we know
    $$V_k^{n+1} = e^{-r\triangle t}[qV_k^{n+1} + (1-q)V_{k+1}^{n+1}]$$

Pricing path-dependent options in a binomial tree

- Recall that in a binomial tree:
  - We set

    $$s_k^n = S_0 u^{n-k} d^k, \qquad \text{for } n = 0, 1, ..., N \text{ and } k = 0, 1, ..., n$$

    as the possible values of stock price at each time point
  - The transition dynamics of S is known precisely: when $S_n = s_k^n$ moves to $s_{k_{new}}^{n+1}$, then $k_{new} = k$ or $k_{new} = k + 1$
  - For a simple non-path-dependent option, if we denote V_k^n as the time-n option price when the current stock price is s_k^n then we know

    $$V_k^{n+1} = e^{-r\triangle t}[qV_k^{n+1} + (1-q)V_{k+1}^{n+1}]$$

- For a path-dependent option, the option price in general is a function of current time n, current stock price level S_n and current value of F_n

Pricing path-dependent options in a binomial tree

- Recall that in a binomial tree:
  - We set

$$s_k^n = S_0 u^{n-k} d^k, \qquad \text{for } n = 0, 1, ..., N \text{ and } k = 0, 1, ..., n$$

    as the possible values of stock price at each time point
  - The transition dynamics of S is known precisely: when $S_n = s_k^n$ moves to $s_{k_{new}}^{n+1}$, then $k_{new} = k$ or $k_{new} = k + 1$
  - For a simple non-path-dependent option, if we denote V_k^n as the time-n option price when the current stock price is s_k^n then we know

$$V_k^{n+1} = e^{-r\triangle t}[qV_k^{n+1} + (1-q)V_{k+1}^{n+1}]$$

- For a path-dependent option, the option price in general is a function of current time n, current stock price level S_n and current value of F_n
- Let $V_{k,j}^n$ be the time-n option price when $S_n = s_k^n$ and $F_n = f_j^n$. Then there are **two questions we must address**:
  - We need to define f_j^n which represents the grid of values to be taken by the process F
  - We need to understand how j the state index of F moves as $s_k^n \to s_{k_{new}}^{n+1}$:

$$V_{k,j}^n = e^{-r\triangle t}[qV_{k,?}^{n+1} + (1-q)V_{k+1,?}^{n+1}]$$

The forward shooting grid method

The pricing strategy of a path-dependent option under a tree model roughly goes as follows:

1. Pick a tree model of the stock price
2. Pick a suitable auxiliary process $F = (F_n)$ based on the nature of the path-dependent option
3. Specify a grid of values f_j^n to contain the possible values of the auxiliary variable F_n at each time point
4. Study how the auxiliary variable moves as the stock price changes, and identify how the indexing system should be updated
5. Write down the recursive equation for the fair option price (with interpolation adjustment if needed)

We demonstrate how these procedures are implemented in two concrete examples.

Lookback option

- Lookback option is a derivative product which payoff function is directly linked to the maximum or minimum stock price level over the product's lifetime. The variations include:

Type of lookback option	Payoff function: Discrete model	Payoff function: Continuous model
Fixed strike lookback call	$\left(\max_{i=0,1,\dots,N} S_i - K\right)^+$	$\left(\sup_{0\leq t\leq T} S_t - K\right)^+$
Fixed strike lookback put	$\left(K - \min_{i=0,1,\dots,N} S_i\right)^+$	$\left(K - \inf_{0\leq t\leq T} S_t\right)^+$
Floating strike lookback call	$\left(S_N - \min_{i=0,1,\dots,N} S_i\right)$	$\left(S_T - \inf_{0\leq t\leq T} S_t\right)$
Floating strike lookback put	$\left(\max_{i=0,1,\dots,N} S_i - S_N\right)$	$\left(\sup_{0\leq t\leq T} S_t - S_T\right)$

Pricing floating strike lookback put with binomial tree

We use an N-period binomial tree with Cox-Ross-Rubinstein (CRR) parameterisation to price a floating strike lookback put option with payoff

$$\left(\max_{i=0,\dots,N} S_i - S_N \right).$$

Suppose the tree parameters $(u, d = 1/u, q)$ have been determined accordingly.

1. **Fix a tree model of stock price**:
   - We are using a binomial tree model with CRR specification such that $ud = 1$. Then the possible node values of the stock price are

$$s_k^n = S_0 u^{n-k} d^k = S_0 u^{n-2k} \quad \text{for } n = 0, 1, \dots, N \text{ and } k = 0, \dots, n$$

2. **Pick an auxiliary variable**:
   - Since we are working with lookback option involving the maximum value of the stock price path, we choose $M_n := \max_{i=0,\dots,n} S_i$ representing the running maximum of the stock price between time zero and time n. The payoff function becomes

$$g(S_N, M_N) := M_N - S_N$$

Pricing floating strike lookback put with binomial tree (cont')

3. **Specify a grid of values for the auxiliary variable M_n:**
   - Since the initial stock price is S_0 and the largest possible price at time n is $S_0 u^n$, the maximum price recorded between time zero and time n must take value in the set

$$\{S_0, S_0 u, S_0 u^2, ..., S_0 u^n\}$$

   - Hence we construct the grid for M_n as

$$m_j^n = S_0 u^{n-j} \quad \text{for } n = 0, 1, ..., N \text{ and } \quad j = 0, 1, ..., n$$

   which represents the j-th possible value of M_n

Pricing floating strike lookback put with binomial tree (cont')

4. **Evolution of auxiliary variable as stock price changes**:
   - As M_{n+1} is the running maximum from time zero to time $n+1$, we expect
   $$M_{n+1} = \max(M_n, S_{n+1}) \tag{2.2}$$
   - We need to translate (2.2) to our indexing system to understand how M_n evolves from m_j^n as $s_k^n \to s_{k_{new}}^{n+1}$
   - Using (2.2) and the definition that $s_k^n = S_0 u^{n-2k}$ and $m_j^n = S_0 u^{n-j}$, we have
   $$\begin{aligned} m_{j_{new}}^{n+1} &= \max(m_j^n, s_{k_{new}}^{n+1}) \\ \implies \quad S_0 u^{n+1-j_{new}} &= \max(S_0 u^{n-j}, S_0 u^{(n+1)-2k_{new}}) \\ \implies \quad n+1-j_{new} &= \max(n-j, n+1-2k_{new}) \\ \implies \quad j_{new} &= n+1-\max(n-j, n+1-2k_{new}) \\ &= \min(2k_{new}, j+1) \\ &=: \phi(k_{new}, j) \end{aligned} \tag{2.3}$$
   - $\phi(k_{new}, j)$ is called the shooting function which describes the new state of the auxiliary variable M when stock price moves to a new state k_{new} and that the current state of M is j

Pricing floating strike lookback put with binomial tree (cont')

5. **Write down the recursive equation for the fair option prices**:
   - If $V^n_{k,j}$ represents the time-n option value when the stock price is $S_n = s^n_k$ and the running maximum is $M_n = m^n_j$, then

     $$V^n_{k,j} = e^{-r\triangle t}[qV^{n+1}_{k,\phi(k,j)} + (1-q)V^{n+1}_{k+1,\phi(k+1,j)}]$$

     where $\phi(\cdot,\cdot)$ is given by (2.3)
   - Note that it is not necessary to compute $V^n_{k,j}$ for all combinations of (k,j). Since by definition we expect $M_n \geq S_n$, we only need to consider the combinations of (k,j) such that

     $$m^n_j \geq s^n_k \implies S_0 u^{n-j} \geq S_0 u^{n-2k} \implies j \leq 2k$$

Pricing floating strike lookback put: a summary

The complete algorithm is as follows:

1. Define $s_k^n = S_0 u^{n-2k}$ and $m_j^n = S_0 u^{n-j}$ for $n = 0, 1, ..., N$ and $j, k = 0, 1, ...n$
2. Option price at terminal time $n = N$ is given by the payoff function:
   - For each $k = 0, 1, ..., N$
     - ★ For each $j = 0, 1, ..., \min(2k, N)$, compute

$$V_{k,j}^N = g(s_k^N, m_j^N) = m_j^N - s_k^N$$

3. Loop backward in time. For $n = N-1, N-2, ..., 0$:
   - For each $k = 0, 1, ..., n$
     - ★ For each $j = 0, 1, ..., \min(2k, n)$

$$V_{k,j}^n = e^{-r\triangle t}[qV_{k,\phi(k,j)}^{n+1} + (1-q)V_{k+1,\phi(k+1,j)}^{n+1}]$$

   where the shooting function is $\phi(k_{new}, j) := \min(2k_{new}, j+1)$ as derived in (2.3) previously
4. The required time-zero option value is $V_{0,0}^0$

Pricing floating strike lookback put: numerical example

Use a two-period ($N = 2$) CRR binomial tree to price a floating strike lookback put option with maturity of $T = 1$ year. Other parameters are: $S_0 = 100$, $r = 1\%$, $\sigma = 20\%$.

- Compute the CRR tree parameters as

$$\triangle t = \frac{T}{N} = 0.5, \quad u = e^{\sigma\sqrt{\triangle t}} = 1.15, \quad , d = \frac{1}{u} = 0.87, \quad q = \frac{e^{r\triangle t} - d}{u - d} = 0.48$$

- Grid of possible stock price values $s_k^n = S_0 u^{n-k} d^k$ are given by

		n = 0	1	2
j	0	100	115.19	132.69
	1		86.81	100
	2			75.36

- Grid of possible running maximum values $m_j^n = S_0 u^{n-j}$ are given by

		n = 0	1	2
j	0	100	115.19	132.69
	1		100	115.19
	2			100

Pricing floating strike lookback put: numerical example (cont')

- Option values at terminal time $n = 2$ are given by the payoff function such that $V_{k,j}^2 = m_j^2 - s_k^2$. Here all values of $V_{k,j}^2$ can be computed as

j	m_j^2	k / s_k^2	0 / 132.69	1 / 100	2 / 75.36
0	132.69		0	32.69	57.33
1	115.19			15.19	39.83
2	100			0	24.64

  Recall it is sufficient to consider the combination of (k, j) such that $m_j^2 \geq s_k^2 \implies j \leq 2k$

- Backward induction: the recursive equation of option price is given by

$$V_{k,j}^n = e^{-r\triangle t}[qV_{k,\phi(k,j)}^{n+1} + (1-q)V_{k+1,\phi(k+1,j)}^{n+1}]$$

  for $k = 0, 1, ..., n$ and $j = 0, 1, ..., \min(2k, n)$. Recall that $\phi(k_{new}, j) = \min(2k_{new}, j+1)$

- As an example say we want to compute $V_{0,0}^1$ (i.e. the option value at time $n = 1$ when $S_1 = s_0^1 = 115.19$ and $M_1 = m_0^1 = 115.19$), then

$$\phi(0,0) = \min(0,1) = 0, \qquad \phi(1,0) = \min(2,1) = 1$$

  and thus

$$\begin{aligned} V_{0,0}^1 &= e^{-r\triangle t}[qV_{0,\phi(0,0)}^2 + (1-q)V_{1,\phi(1,0)}^2] \\ &= e^{-r\triangle t}[qV_{0,0}^2 + (1-q)V_{1,1}^2] = 7.82 \end{aligned}$$

Pricing floating strike lookback put: numerical example (cont')

- We can compute the values of $V^1_{1,0}$, and $V^1_{1,1}$ similarly where the results are

j	m^1_j	k / s^1_k	0 / 115.19	1 / 86.81
0	115.19		7.82	27.80
1	100			12.69

- Finally, the fair option value at time zero is

$$
\begin{aligned}
V^0_{0,0} &= e^{-r\triangle t}[qV^1_{0,\phi(0,0)} + (1-q)V^1_{1,\phi(1,0)}] \\
&= e^{-r\triangle t}[qV^1_{0,0} + (1-q)V^1_{1,1}] \\
&= 10.29
\end{aligned}
$$

Asian option

- Asian option is a derivative product which payoff function depends on the arithmetic average of the stock prices over the product's lifetime. The variations include:

Type of Asian option	Payoff function: Discrete model	Payoff function: Continuous model
Fixed strike Asian call	$\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^+$	$\left(\frac{1}{T}\int_0^T S_u du - K\right)^+$
Fixed strike Asian put	$\left(K - \frac{1}{N+1}\sum_{i=0}^{N} S_i\right)^+$	$\left(K - \frac{1}{T}\int_0^T S_u du\right)^+$
Floating strike Asian call	$\left(S_N - \frac{1}{N+1}\sum_{i=0}^{N} S_i\right)^+$	$\left(S_T - \frac{1}{T}\int_0^T S_u du\right)^+$
Floating strike Asian put	$\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - S_N\right)^+$	$\left(\frac{1}{T}\int_0^T S_u du - S_T\right)^+$

Pricing fixed strike Asian call with binomial tree

We use an N-period binomial tree with CRR parameterisation to price a fixed strike Asian call option with payoff

$$\left(\frac{1}{N+1}\sum_{i=0}^{N} S_i - K\right)^{+}.$$

Suppose the tree parameters $(u, d = 1/u, q)$ have been determined accordingly.

1. **Fix a tree model of stock price**:
   - Again we are using a binomial tree model with CRR specification with $ud = 1$. The possible node values of the stock price are

     $$s_k^n = S_0 u^{n-k} d^k = S_0 u^{n-2k} \quad \text{for } n = 0, 1, ..., N \text{ and } k = 0, ..., n$$

2. **Pick an auxiliary variable**:
   - Take $A_n := \frac{1}{n+1}\sum_{i=0}^{n} S_i$ which represents the running average of the stock price up to time n. The payoff function becomes

     $$g(A_N) := (A_N - K)^{+}$$

Pricing fixed strike Asian call with binomial tree (cont')

3. **Specify a grid of values for the auxiliary variable A_n:**
   - It is computational infeasible to construct a grid covering all possible values of A_n because in general the total number of possible values at time n is 2^n
   - Hence we use a parameterised grid instead:
     - ★ Fix $\rho > 0$. Define

$$a_j^n = S_0 u^{\rho j}, \qquad -J_{min}^n \leq j \leq J_{max}^n$$

     - ★ We are going to choose the positive integers J_{min}^n and J_{max}^n just large enough to ensure the a_j^n's cover the possible range of A_n
     - ★ Note that $\ln a_j^n$'s are evenly spaced: $\ln a_{j+1}^n - \ln a_j^n = \rho \ln u$
     - ★ ρ is called the quantisation parameter

Pricing fixed strike Asian call with binomial tree (cont')

3. **Specify a grid of values for the auxiliary variable A_n (cont')**:
   - At $n = 0$, $S_0 = A_0$ so we can pick $J^0_{max} = J^0_{min} = 0$
   - Observe that

$$A_n = \frac{1}{n+1}\sum_{i=0}^{n} S_i = \frac{1}{n+1}\left(\sum_{i=0}^{n-1} S_i + S_n\right) = \frac{nA_{n-1} + S_n}{n+1}$$

   - Suppose J^{n-1}_{max} has been fixed. Then since the largest possible value of S_n is $S_0 u^n$, the largest possible value of A_n will be

$$a^n_{max} = \frac{nS_0 u^{\rho J^{n-1}_{max}} + S_0 u^n}{n+1}$$

   - We want to find the smallest possible J^n_{max} such that $a^n_{max} \leq S_0 u^{\rho J^n_{max}}$. We hence pick

$$J^n_{max} = \left\lceil \frac{1}{\rho \ln u} \ln \frac{a^n_{max}}{S_0} \right\rceil = \left\lceil \frac{1}{\rho \ln u} \ln \frac{nu^{\rho J^{n-1}_{max}} + u^n}{n+1} \right\rceil \tag{2.4}$$

     where $\lceil \cdot \rceil$ denotes the ceiling of a number. With $J^0_{max} = 0$, (2.4) inductively defines J^n_{max} for all n
   - With similar logic, we choose

$$J^n_{min} = \left\lceil \frac{-1}{\rho \ln u} \ln \frac{nu^{-\rho J^{n-1}_{min}} + d^n}{n+1} \right\rceil \tag{2.5}$$

Pricing fixed strike Asian call with binomial tree (cont')

4. **Evolution of auxiliary variable as stock price changes**:
   - The transition function of A_n is

$$A_{n+1} = \frac{(n+1)A_n + S_{n+1}}{n+2} \tag{2.6}$$

   We need to translate (2.6) to our indexing system to understand how A_n evolves from a_j^n as $s_k^n \to s_{k_{new}}^{n+1}$

   - Using (2.6) and the definition that $s_k^n = S_0 u^{n-2k}$ and $a_j^n = S_0 u^{\rho j}$, we have

$$\begin{aligned} S_0 u^{\rho j_{new}} &= \frac{(n+1)S_0 u^{\rho j} + S_0 u^{n+1-2k_{new}}}{n+2} \\ \implies j_{new} &= \frac{1}{\rho \ln u} \ln \frac{(n+1)u^{\rho j} + u^{n+1-2k_{new}}}{n+2} \\ &=: \phi(n, k_{new}, j) \end{aligned} \tag{2.7}$$

   - The problem is that $\phi(n, k_{new}, j)$ may not give a proper integer! This happens when the predetermined grid of a_j^{n+1} does not contain the projected value of A_{n+1}

Pricing fixed strike Asian call with binomial tree (cont')

5. **Write down the recursive equation of the fair option prices**:
   - The recursive equation is supposed to be
     $$V^n_{k,j} = e^{-r\triangle t}[qV^{n+1}_{k,\phi(n,k,j)} + (1-q)V^{n+1}_{k+1,\phi(n,k+1,j)}]$$
     but of course the above is not well-defined if $\phi(\cdot,\cdot,\cdot)$ is not an integer
   - Recall that $V^{n+1}_{k_{new},\phi(n,k_{new},j)}$ is supposed to give the option value when $S_{n+1} = s^{n+1}_{k_{new}}$ and
     $$A_{n+1} = a^{n+1}_{\phi(n,k_{new},j)} = \frac{(n+1)a^n_j + s^{n+1}_{k_{new}}}{n+2}.$$
     We will estimate $V^{n+1}_{k_{new},\phi(n,k_{new},j)}$ by linear interpolation on the grid of $\ln a^{n+1}_j$
   - Let $P[\cdot]$ denote the linear interpolation operator such that
     $$P[V^{n+1}_{k_{new},\phi(n,k_{new},j)}] := \alpha V^{n+1}_{k_{new},\phi_-(n,k_{new},j)} + (1-\alpha)V^{n+1}_{k_{new},\phi_+(n,k_{new},j)}$$
     with
     $$\phi_+(n,k_{new},j) := \lceil \phi(n,k_{new},j) \rceil, \qquad \phi_-(n,k_{new},j) := \lfloor \phi(n,k_{new},j) \rfloor$$
     and
     $$\alpha := \frac{\ln a^{n+1}_{\phi_+(n,k_{new},j)} - \ln a^{n+1}_{\phi(n,k_{new},j)}}{\ln a^{n+1}_{\phi_+(n,k_{new},j)} - \ln a^{n+1}_{\phi_-(n,k_{new},j)}} = \frac{\ln a^{n+1}_{\phi_+(n,k_{new},j)} - \ln a^{n+1}_{\phi(n,k_{new},j)}}{\rho \ln u}.$$
     Here $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the ceiling and floor of a number respectively
   - The modified recursion is
     $$V^n_{k,j} = e^{-r\triangle t}\left\{qP[V^{n+1}_{k,\phi(n,k,j)}] + (1-q)P[V^{n+1}_{k+1,\phi(n,k+1,j)}]\right\}$$

Pricing fixed strike Asian call: numerical example

Use a three-period ($N = 3$) CRR binomial tree to price a fixed strike Asian call option of maturity $T = 1$ year. Other parameters are: $S_0 = 100$, $K = 100$, $r = 1\%$, $\sigma = 20\%$. Use $\rho = 0.5$ for the grid of the running averages.

- The CRR tree parameters are given by

$$\triangle t = \frac{T}{N} = 0.33, \quad u = e^{\sigma\sqrt{\triangle t}} = 1.12, \quad , d = \frac{1}{u} = 0.89, \quad q = \frac{e^{r\triangle t} - d}{u - d} = 0.49$$

- Stock price grid values are given by $s_k^n = S_0 u^{n-k} d^k$:

		n			
		0	1	2	3
	0	100	112.24	125.98	141.40
k	1		89.09	100	112.24
	2			79.38	89.09
	3				70.72

- Use (2.4) and (2.5) to work out the values of J_{max}^n and J_{min}^n for the construction of grid for A_n:

	n			
	0	1	2	3
J_{max}^n	0	2	3	4
J_{min}^n	0	1	2	3

Pricing fixed strike Asian call: numerical example (cont')

- The grid of A_n is constructed as $a_j^n = S_0 u^{\rho j}$ for $-J_{min}^n \leq j \leq J_{max}^n$

j	$n=0$	1	2	3
4				125.98
3			118.91	118.91
2		112.24	112.24	112.24
1		105.94	105.94	105.94
0	100	100	100	100
-1		94.39	94.39	94.39
-2			89.09	89.09
-3				84.10

- The terminal option price at $n = 3$ is $V_{k,j}^3 = (a_j^3 - K)^+$. Hence for all $k = 0, 1, 2, 3$ we have

j	4	3	2	1	0	-1	-2	-3
a_j^3	125.98	118.91	112.24	105.94	100	94.39	89.09	84.10
$V_{k,j}^3$	25.98	18.91	12.24	5.94	0.00	0.00	0.00	0.00

Pricing fixed strike Asian call: numerical example (cont')

- Suppose we want to compute $V^2_{1,0}$ which is the option value at time $n = 2$ when the current stock price is $s^2_1 = 100$ and the current running average is $a^2_0 = 100$
- Recall the transition dynamics $A_{n+1} = \frac{(n+1)A_n + S_{n+1}}{n+2}$. There are two cases:
  - If the stock goes up such that $S_3 = s^3_1 = 112.24$, then $A_3 = \frac{3\times 100+112.24}{4} = 103.06$
  - If the stock goes down such that $S_3 = s^3_2 = 89.09$, then $A_3 = \frac{3\times 100+89.09}{4} = 97.27$
  - Neither of these values lie on the grid of a^3_j
- The standard recursive equation is supposed to be

$$V^n_{k,j} = e^{-r\triangle t}[qV^{n+1}_{k,\phi(n,k,j)} + (1-q)V^{n+1}_{k+1,\phi(n,k+1,j)}]$$

where $\phi(n, k_{new}, j)$ in defined in (2.7). We have

$$\phi(2,1,0) = 0.52, \qquad \phi(2,2,0) = -0.48$$

and thus

$$V^2_{1,0} = e^{-r\triangle t}[qV^3_{1,0.52} + (1-q)V^3_{2,-0.48}]$$

- We need to use interpolation to approximate $V^3_{1,0.52}$ and $V^3_{2,-0.47}$

Pricing fixed strike Asian call: numerical example (cont')

- To approximate $V^3_{1,0.52}$, we look at the nearest available option values:

	j = 1	0.52	0
a^3_j	105.94	103.06	100.00
$\ln a^3_j$	4.66	4.64	4.61
$V^3_{1,j}$	5.94	?	0.00

Using linear interpolation in $\ln a^3_j$, we compute

$$\begin{aligned} P[V^3_{1,0.52}] &= \alpha V^3_{1,\phi_-(2,1,0)} + (1-\alpha) V^3_{1,\phi_+(2,1,0)} \\ &= \frac{\ln a^3_1 - \ln a^3_{0.52}}{\rho \ln u} V^3_{1,0} + \left(1 - \frac{\ln a^3_1 - \ln a^3_{0.52}}{\rho \ln u}\right) V^3_{1,1} \\ &= \frac{4.66 - 4.64}{0.05} \times 0 + \frac{4.64 - 4.61}{0.05} \times 5.94 = 3.10 \end{aligned}$$

- By similar interpolation exercise again, we can obtain $P[V^3_{2,-0.47}] = 0$
- This finally gives $V^2_{1,0} \approx e^{-r\triangle t}\{qP[V^3_{1,0.52}] + (1-q)P[V^3_{2,-0.47}]\} = 1.50$
- Ultimately we can get $V^0_{0,0} = 4.81$

American path-dependent option

- There also exists American style of path-dependent option where the option holder can freely exercise the option at any stopping time τ to receive the payoff $g(S_0, S_1, ..., S_\tau)$ immediately

- If the payoff function can be converted into $g(S_\tau, F_\tau)$ for some auxiliary process F, then the time-n fair value of the option is

$$\sup_{\tau \in \mathcal{T}_{n,N}} \mathbb{E}_{\mathbb{Q}} \left[e^{-r(\tau - n)\triangle t} g(S_\tau, F_\tau) \middle| \mathcal{F}_n \right]$$

  where $\mathcal{T}_{n,N}$ is the set of stopping times taking values in $\{n, n+1, ..., N\}$

- Just like Prop 1.6 in Topic 1, pricing of the American version of the path-dependent option is very easy in a lattice model where we just need to consider the maximum of intrinsic value and continuation value of the option in the backward induction procedure

- E.g. in a binomial tree model the recursive equation will become

$$V_{k,j}^n = \max \left\{ g(s_k^n, f_j^n), e^{-r\triangle t} \left[qV_{k,\phi(n,k,j)}^{n+1} + (1-q)V_{k+1,\phi(n,k+1,j)}^{n+1} \right] \right\}$$

  where $g(s_k^n, f_j^n)$ is the intrinsic value of the option and $\phi(\cdot,\cdot,\cdot)$ is some suitably constructed shooting function

- The ability to handle American path-dependent options with ease is the most powerful feature of a lattice method

Optional reading

- Barraquand, J. & Pudet, T. (1996). Pricing of American Path-dependent Contingent Claims. Mathematical Finance, 6(1), 17 - 51.
- Hull, J. & White A. (1993). Efficient Procedures for Valuing European and American Path Dependent Options. Journal of Derivatives, 1, 21 - 31.
- Forsyth, P. A., Vetzal, K. R. & Zvan, R. (2002). Convergence of Numerical Methods for Valuing Path-dependent Options Using Interpolation. Review of Derivatives Research, 5(3), 273 - 314.