Chapter 3

Multi-period Binomial Tree Method

For any time interval $[t, t+\Delta t]$, assume S goes either up to Su or down to Sd, and correpondingly V goes to V_u or V_d. Then we have

$$V = e^{-r\Delta t}\left[pV_u + (1-p)\,V_d\right],$$

where

$$p = \frac{e^{r\Delta t} - d}{u - d}.$$

3.1 Multi-period BTM for Vanilla Options

3.1.1 European Options

Let T be expiration date, $[0, T]$ be the lifetime of an European vanilla option. If N is the number of discrete time points, we have time points $n\Delta t$, $n = 0, 1, ..., N$, with $\Delta t = \frac{T}{N}$. At time $t = 0$, the underlying asset price is known, denoted by S_0. At time Δt, there are two possible underlying asset prices, $S_0 u$ and $S_0 d$. Without loss of generality, we assume $ud = 1$. At time $2\Delta t$, there are three possible underlying asset prices, $S_0 u^2$, S_0, and $S_0 d^2 = S_0 u^{-2}$; and so on. In general, at time $n\Delta t$, $n+1$ underlying asset prices are considered. These are $S_0 u^{-n}, S_0 u^{-n+2}, ..., S_0 u^n$. A complete tree is then constructed. Let V_j^n be the option price at time point $n\Delta t$ with underlying asset price $S_j = S_0 u^j$. Note that S_j will jump either up to S_{j+1} or down to S_{j-1} at time $(n+1)\Delta t$, and the value of the option at $(n+1)\Delta t$ will become either V_{j+1}^{n+1} or V_{j-1}^{n+1}. Since the length of time period is Δt, the discounting factor is $e^{-r\Delta t}$. Then, similar to the arguments in the single-period case, we have

$$V_j^n = e^{-r\Delta t}\left[pV_{j+1}^{n+1} + (1-p)V_{j-1}^{n+1}\right],\ j = -n, -n+2, ..., n,\ n = 0, 1, ..., N-1.$$

At expiry,

$$V_j^N = \begin{cases} (S_0 u^j - K)^+ & \text{for call,} \\ (K - S_0 u^j)^+ & \text{for put,} \end{cases} \quad j = -N, -N+2, ..., N.$$

This is the multi-period binomial tree method (BTM).

For simplicity, we can write the BTM for European vanilla options as

$$V(S, t-\Delta t) = e^{-r\Delta t}\left[pV(Su, t) + (1-p)V(Sd, t)\right],$$

with the terminal condition

$$V(S,T) = \varphi(S) = \begin{cases} (S-K)^+ & \text{for call,} \\ (K-S)^+ & \text{for put,} \end{cases}$$

for all $S > 0$, $t \in [0,T)$.

3.1.2 Continuous-dividend payment

Assume that the underlying asset pays a continuous dividend at yield q. Then, it is not hard to show that the risk-neutral probability

$$p = \frac{e^{(r-q)\Delta t} - d}{u-d}.$$

I leave it as an exercise.

3.1.3 American Options

At any nodes of the tree, we need to check if early exercise is optimal. So, the BTM can be written as

$$V(S,t-\Delta t) = \max\left\{e^{-r\Delta t}\left[pV(Su,t) + (1-p)V(Sd,t)\right], \varphi(S)\right\}$$

with terminal condition $V(S,T) = \varphi(S)$, for all $S > 0$, $t \in [0,T)$.

3.2 Multi-period BTM for Exotic Options

3.2.1 Barrier Options

Consider an up-out option: if the barrier H is hit ($H > S_0$), then the option does not pay off. The BTM becomes

$$V(S,t-\Delta t) = e^{-r\Delta t}\left[pV(Su,t) + (1-p)V(Sd,t)\right]$$

and

$$V(H,t) = 0,$$

with terminal condition $V(S,T) = \varphi(S)$, for all $0 < S < H$, $t \in [0,T)$.

In general, we cannot ensure that the line $S = H$ coincides with the tree if we start for a given initial stock price S_0. Then we have to take an approximation: the option takes zero value at the tree nodes closest to $S = H$, which would spoil accuracy.

To cure the problem, it is better to use a finite difference method to be introduced later (or a trinomial tree method). If we stick to a BTM, it is unnecessary for a tree to start from the given S_0, and we can make use of interpolation of the option prices computed from a tree that coincides with $S = H$.

3.2.2 Lookback Options

Let N be the number of discrete time points and we have time points $t_n = n\Delta t$, $n = 0, 1, \ldots, N$ with $\Delta t = T/N$. Denote by S_i the underlying asset value of a path at time t_i, $i = 0, 1, \cdots, n$ and introduce a path-dependent variable associated with time t_n

$$A_n = \begin{cases} \max\limits_{1\le i\le n} S_i, & \text{lookback max} \\ \min\limits_{1\le i\le n} S_i, & \text{lookback min} \end{cases}$$

Consider the lookback options with the following payoffs:

$$\Lambda(S_N, A_N) = \begin{cases} (A_N - K)^+\,, & \text{lookback call} \\ (K - A_N)^+\,, & \text{lookback put} \end{cases} \quad \text{(fixed strike)}$$

and

$$\Lambda(S_N, A_N) = \begin{cases} S_N - A_N, & \text{lookback call} \\ A_N - S_N, & \text{lookback put} \end{cases} \quad \text{(floating strike)}$$

At time $t = t_n$, the option value depends on the path-dependent variable A_n in addition to S_n and t_n. It is assumed that S_n will either jump up to $S_n u$ with probability p or down to $S_n d$ with probability $1 - p$ at time t_{n+1}. Consequently,

$$A^u_{n+1} = \begin{cases} \max(A_n, S_n u), & \text{lookback max} \\ A, & \text{lookback min} \end{cases} \quad \text{for up movement}$$

and

$$A^d_{n+1} = \begin{cases} A_n, & \text{lookback max} \\ \min\left(A_n, S_n d\right), & \text{lookback min} \end{cases} \quad \text{for down movement}$$

By no-arbitrage argument, one has

$$V^n(S_n, A_n) = e^{-r\Delta t}[pV^{n+1}(S_n u, A^u_{n+1}) + (1-p)V^{n+1}(S_n d, A^d_{n+1})]. \tag{3.1}$$

where $p = \frac{e^{r\Delta t} - d}{u - d}$, $u = e^{\sigma\sqrt{\Delta t}}$, $d = e^{-\sigma\sqrt{\Delta t}}$. At expiry (i.e. $T = N\Delta t$) we have

$$V^N(S_N, A_N) = \Lambda(S_N, A_N). \tag{3.2}$$

Using the backward procedure (3.1)-(3.2), option prices can be calculated.

Note that (3.1)-(3.2) can be rewritten as

$$\begin{aligned} V(S, A, t - \Delta t) &= e^{-r\Delta t}[pV(Su, A^u, t) + (1-p)V(Sd, A^d, t)], & (3.3) \\ V(S, A, T) &= \Lambda(S, A), & (3.4) \end{aligned}$$

in $t \in [0, T)$, $S > A$ for lookback min, $S < A$ for lookback max, where

$$A^u = \begin{cases} \max(A, Su), & \text{lookback max} \\ A, & \text{lookback min} \end{cases}$$

and

$$A^d = \begin{cases} A, & \text{lookback max} \\ \min\left(A, Sd\right), & \text{lookback min} \end{cases}$$

3.2.3 Asian options

Now let us turn to the case of Asian options, where the following path-dependent variable, still denote by A, is introduced.

$$A_n = \begin{cases} \frac{1}{n}\Sigma_{i=1}^n S_i, \text{ Asian arithmetic} \\ (\Pi_{i=1}^n S_i)^{1/n}, \text{ Asian geometric} \end{cases}$$

Consider the Asian options with payoffs:

$$\Lambda(S_N, A_N) = \begin{cases} (A_N - K)^+, \text{ Asian call} \\ (K - A_N)^+, \text{ Asian put} \end{cases} \text{ (fixed strike)}$$

and

$$\Lambda(S_N, A_N) = \begin{cases} (S_N - A_N)^+, \text{ Asian call} \\ (A_N - S_N)^+, \text{ Asian put} \end{cases} \text{ (floating strike)}$$

At time $t = t_n$, the option value $V^n = V^n(S_n, A_n)$. It is easy to see that (3.1)-(3.2) still hold for Asian options, where

$$A_{n+1}^u = \begin{cases} \frac{nA_n+S_nu}{n+1}, \text{ Asian arithmetic} \\ (A_n^n S_n u)^{1/(n+1)}, \text{ Asian geometric} \end{cases} \text{ for up movement}$$

and

$$A_{n+1}^d = \begin{cases} \frac{nA_n+S_nd}{n+1}, \text{ Asian arithmetic} \\ (A_n^n S_n d)^{1/(n+1)}, \text{ Asian geometric} \end{cases} \text{ for down movement}$$

Similarly, we have (3.3)-(3.4) in $t \in [0, T)$, $S > 0$, $A > 0$, where

$$A^u = \begin{cases} \frac{(t-\Delta t)A+Su\Delta t}{t}, \text{ Asian arithmetic} \\ A^{\frac{t-\Delta t}{t}}(Su)^{\frac{\Delta t}{t}}, \text{ Asian geometric} \end{cases}$$

and

$$A^d = \begin{cases} \frac{(t-\Delta t)A+Sd\Delta t}{t}, \text{ Asian arithmetic} \\ A^{\frac{t-\Delta t}{t}}(Sd)^{\frac{\Delta t}{t}}, \text{ Asian geometric} \end{cases}$$

3.3 Single-state Variable BTM for Floating Strike Lookback/Asian Options

For the floating strike payoff, it is not hard to verify

$$V(\lambda S, \lambda A, t) = \lambda V(S, A, t).$$

It follows by taking $\lambda = \frac{1}{S}$ that

$$\frac{1}{S}V(S, A, t) = V(1, \frac{A}{S}, t) \equiv W(x, t) \text{ and } x = \frac{A}{S}.$$

Then (3.3) reduces to

$$\begin{aligned} SW(x, t-\Delta t) &= e^{-r\Delta t}[pSuW(\frac{A^u}{Su}, t) + (1-p)SdW(\frac{A^d}{Sd}, t)] \\ &= e^{-r\Delta t}[pSuW(x_u, t) + (1-p)SdW(x_d, t)], \end{aligned}$$

namely,

$$W(x, t-\Delta t) = e^{-r\Delta t}[puW(x_u, t) + (1-p)dW(x_d, t)],$$

where

$$x_u = \begin{cases} \frac{(t-\Delta t)xd+\Delta t}{t}, \text{ Asian arithmetic} \\ (xd)^{\frac{t-\Delta t}{t}}, \text{ Asian geometric} \\ \max(xd, 1), \text{ lookback max} \\ xd, \text{ lookback min} \end{cases}$$

and

$$x_d = \begin{cases} \frac{(t-\Delta t)xu+\Delta t}{t}, \text{ Asian arithmetic} \\ (xu)^{\frac{t-\Delta t}{t}}, \text{ Asian geometric} \\ xu, \text{ lookback max} \\ \min(xu, 1), \text{ lookback min} \end{cases}$$

The terminal condition is

$$W(x, T) = \begin{cases} (x-1)^+ \text{ for floating put} \\ (1-x)^+ \text{ for floating call.} \end{cases}$$

It should be pointed out that $x > 0$ for Asian options, while $x \geq 1$ for lookback max and $x \leq 1$ for lookback min.

Exercise: plot a 3-step single-state variable binomial tree for the lookback max option and the Asian arithmetic option, respectively.

Remark 10 *The above reduction applies to the American-style floating strike Asian/lookback options. For European-style fixed strike case, another reduction is also available but cannot be extended to the American case (see, for example, Dai (2003) One-state variable binomial models for European-/American-style geometric Asian options, Quantitative Finance, 3(4):288-295).*

3.4 Modified BTM: Forward Shooting Grid Method

For Asian arithmetic options, BTM is not feasible since the number of possible arithmetic average values increases exponentially with the number of timesteps. With interpolation technique, a remedy proposed by Hull and White (1993) is to restrict the possible average values to a set of predetermined values. Barraquand and Pudet (1996) present a similar algorithm, known as the forward shooting grid method (FSGM).

Adopting the notation of Barraquand and Pudet (1996), we present the algorithm for Asian arithmetic options. For Δt given, let

$$\Delta Y = \rho\sigma\sqrt{\Delta t}. \tag{3.5}$$

Here ρ is a quantization parameter for spacing in the average direction and $1/\rho$ is assumed to be an integer. Later we will see that, in order to guarantee convergence, ρ also depends on Δt. Let discrete values of the asset price S and the arithmetic average price A be given by

$$S_j^n = u^j \text{ and } A_k^n = e^{k\Delta Y} \tag{3.6}$$

for $n = 0, \ldots, N$ and $j, k \in Z$. It is assumed that (S_j^n, A_k^n) will either jump up to $(S_{j+1}^{n+1}, A_{k^+}^{n+1})$ with probability p or down to $(S_{j-1}^{n+1}, A_{k^-}^{n+1})$ with probability $1-p$, where

$$A_{k^+}^{n+1} = \frac{nA_k^n + S_{j+1}^{n+1}}{n+1}\ ,\ A_{k^-}^{n+1} = \frac{nA_k^n + S_{j-1}^{n+1}}{n+1}. \tag{3.7}$$

Note that $A_{k^\pm}^{n+1}$ in general does not coincide with $A_{k'}^{n+1} = e^{k'\Delta Y}$, for some integer k', thus some form of interpolation should be taken. For future reference, define

$$k_{floor}^{\pm} = floor\left(\frac{\ln(A_{k^\pm}^{n+1})}{\Delta Y}\right). \tag{3.8}$$

Here $floor(x)$ denotes the largest integer less than or equal to x.

Let $U^n(S_j^n, A_k^n)$ stand for option values at time $t = n\Delta t, S = S_j^n, A = A_k^n$. The backward procedure of the FSGM for Asian arithmetic option is described as follows:

$$\begin{cases} U^n(S_j^n, A_k^n) = e^{-r\Delta t}[p\Pi_A U^{n+1}(S_{j+1}^{n+1}, A_{k^+}^{n+1}) + (1-p)\Pi_A U^{n+1}(S_{j-1}^{n+1}, A_{k^-}^{n+1})] \\ U^N(S_j^N, A_k^N) = \Lambda(S_j^N, A_k^N) \end{cases} \tag{3.9}$$

for $n = N-1, \ldots, 0$; $j, k \in Z$, where Π_A is the interpolation operator. For example, for nearest lattice point, linear or quadratic interpolations, we can write

$$\begin{aligned} &\Pi_A U^{n+1}(S_{j\pm1}^{n+1}, A_{k^\pm}^{n+1}) \\ =\ &\alpha_{-1}^{\pm}U^{n+1}(S_{j\pm1}^{n+1}, A_{k_{floor}^{\pm}-1}^{n+1}) + \alpha_0^{\pm}U^{n+1}(S_{j\pm1}^{n+1}, A_{k_{floor}^{\pm}}^{n+1}) + \alpha_1^{\pm}U^{n+1}(S_{j\pm1}^{n+1}, A_{k_{floor}^{\pm}+1}^{n+1}), \end{aligned} \tag{3.10}$$

where α's are determined by the type of interpolation used and

$$\alpha_{-1}^{\pm} + \alpha_0^{\pm} + \alpha_1^{\pm} = 1. \tag{3.11}$$

Interpolation formulas:

(i) Given $x_0 < x_1$, $f(x_0)$ and $f(x_1)$, the linear interpolation is

$$f(x) = \frac{x - x_0}{x_1 - x_0}f(x_1) + \frac{x_1 - x}{x_1 - x_0}f(x_0).$$

(ii) Given $x_0 < x_1 < x_2$, $f(x_0)$, $f(x_1)$ and $f(x_2)$, the quadratic interpolation is

$$f(x) = \frac{(x_1 - x)(x_2 - x)}{(x_1 - x_0)(x_2 - x_0)}f(x_0) + \frac{(x - x_0)(x_2 - x)}{(x_1 - x_0)(x_2 - x_1)}f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}f(x_2).$$