CHAPTER 6
Numerical Schemes for Pricing Options

In previous chapters, closed form price formulas for a variety of option models have been obtained. However, option models which lend themselves to a closed form price formula are limited. Frequently, option valuation must be resorted to numerical procedures. The common numerical methods employed in option valuation include the *lattice tree methods, finite difference algorithms* and *Monte Carlo simulation*.

The binomial schemes are most widely used in the finance community for numerical valuation of a wide variety of option models, due primarily to its ease of implementation and pedagogical appeal. The primary essence of the binomial model is the simulation of the continuous asset price movement by a discrete random walk model. Interestingly, the concept of risk neutral valuation is imbedded naturally in the binomial model. In Sec. 6.1, we revisit the binomial model and illustrate how to apply the binomial scheme for valuation of options on discrete-dividend paying asset and options with early exercise right and callable right. The asymptotic limit of the discrete binomial model to the continuous Black-Scholes model is examined. We also consider the extension of the binomial lattice to the trinomial lattice. The trinomial tree simulates the underlying asset price process using a discrete three-jump process. The forward shooting grid approach allows us to keep track of path dependent state variables in a lattice tree. We examine how to use such technique to price options with Parisian variant of knock-out feature and Asian options.

The finite difference approach seeks the discretization of the differential operators in the continuous Black-Scholes model. The numerical schemes arising from the discretization procedure can be broadly classified as either implicit or explicit schemes. Each class of schemes have their merits and limitations. The lattice tree schemes can be considered as explicit finite difference schemes, though they are derived using quite different approaches. In Sec. 6.2, various versions of finite difference schemes for option valuation are presented. In particular, we discuss the projected successive-over-relaxation scheme and the front-fixing method for numerical valuation of American options.

Nowadays, it is quite common to demand the computation of thousands of option values within a short duration of time, thus providing the impetus for developing numerical algorithms that compete favorably in terms of accuracy, efficiency and reliability. The theoretical concepts of order of accuracy and numerical stability in the analysis of a numerical scheme are discussed. We analyze the intricacies associated with the smoothing of the "kink" or "jump" in the terminal payoff function and the avoidance of spurious oscillations. Also, the issues of implementing the boundary conditions in barrier option and lookback option are discussed.

The Monte Carlo method simulates the random movement of the asset prices and provides a probabilistic solution to the option pricing models. Since most derivative pricing problems can be formulated as computation of the discounted expectation of the terminal payoff function, the Monte Carlo simulation provides a direct numerical tool for pricing derivative securities, even without a deep understanding of the nature of the pricing model. When faced with pricing of a new derivative with complex payoffs, a market practitioner can always rely on the Monte Carlo simulation procedure to generate an estimate of the price of the new derivative, though other more efficient numerical methods may be available when the analytic properties of the derivative model are better explored.

One main advantage of the Monte Carlo simulation is that it can accommodate without much additional effort complex payoff functions. Also, the computational cost for Monte Carlo simulation increases linearly with the number of underlying state variables, so the method becomes more competitive for multi-state option models with a large number of risky assets. The most undesirable nature of Monte Carlo simulation is that a large number of simulation runs are generally required in order to achieve a desired level of accuracy. This is because the standard error of the estimate is inversely proportional to the square root of the number of simulation runs. To reduce the standard deviation of the estimate, there exist several effective variance reduction techniques, like the control variate technique and the antithetic variables technique. In Sec. 6.3, we examine how to apply these variance reduction techniques in the context of option pricing.

It had been commonly believed that Monte Carlo simulation cannot be used to handle the early exercise decision of an American option since one cannot predict whether the early exercise decision is optimal when the asset price reaches certain level at a particular instant. Recently, several effective Monte Carlo simulation techniques have been proposed for the valuation of American options. These include the bundling and sorting algorithm, the method of parameterization of the optimal exercise boundary, stochastic mesh method and least squares regression method. An account of each of these techniques is presented at the end of Sec. 6.3.

6.1 Lattice tree methods

We start this section by revisiting the binomial model and consider its continuous limits. We then examine how to modify the binomial schemes so as to incorporate the discrete dividend feature, early exercise and call features. Also, we illustrate how to construct the trinomial schemes by equating the mean and variance of the continuous asset price process and its discrete trinomial approximation. At the end of this section, we consider the forward shooting grid approach of pricing path dependent options.

6.1.1 Binomial model revisited

In the discrete binomial pricing model, we simulate the asset price movement by the discrete binomial process. In Sec. 2.1.4, we derive the risk neutral probability $p = \dfrac{R-d}{u-d}$ of upward move in the discrete binomial process. Here, $R = e^{r\Delta t}$ is the growth factor over one period. However, the proportional upward jump u and downward jump d have not yet been determined. We expect u and d to be directly related to the volatility of the continuous diffusion process of the asset price. Such issues are explored as follows.

Let S_t and $S_{t+\triangle t}$ denote, respectively, the asset prices at the current time t and one period $\triangle t$ later. In the Black-Scholes continuous model, the asset price dynamics is assumed to follow the Geometric Brownian motion where $\dfrac{S_{t+\triangle t}}{S_t}$ is lognormally distributed. Under the risk neutral measure, $\ln \dfrac{S_{t+\triangle t}}{S_t}$ becomes normally distributed with mean $\left(r - \dfrac{\sigma^2}{2}\right)\triangle t$ and variance $\sigma^2 \triangle t$ [see Eqs. (2.4.18a,b)], where r is the riskless interest rate and σ^2 is the variance rate. The mean and variance of $\dfrac{S_{t+\triangle t}}{S_t}$ are R and $R^2(e^{\sigma^2\triangle t} - 1)$, respectively [see Eqs. (2.3.23a,b)]. On the other hand, for the one-period binomial option model under the risk neutral measure, the mean and variance of the asset price ratio $\dfrac{S_{t+\triangle t}}{S}$ are

$$pu + (1-p)d \quad \text{and} \quad pu^2 + (1-p)d^2 - [pu + (1-p)d]^2,$$

respectively. By equating the mean and variance of the asset price ratio in both continuous and discrete models, we obtain

$$pu + (1-p)d = R \tag{6.1.1a}$$

$$pu^2 + (1-p)d^2 - R^2 = R^2(e^{\sigma^2\triangle t} - 1). \tag{6.1.1b}$$

Equation (6.1.1a) leads to $p = \dfrac{R-d}{u-d}$, the same risk neutral probability which has been determined in Sec. 2.1.4. Equations (6.1.1a,b) provide only two equations for the three unknowns: u, d and p. The third condition can be chosen arbitrarily. A convenient choice is the tree-symmetry condition

$$u = \frac{1}{d}, \tag{6.1.1c}$$

so that the lattice nodes associated with the binomial tree are symmetrical. Writing $\tilde{\sigma^2} = R^2 e^{\sigma^2 \triangle t}$, the solution to Eqs. (6.1.1a,b,c) is found to be

$$u = \frac{1}{d} = \frac{\tilde{\sigma^2} + 1 + \sqrt{(\tilde{\sigma^2}+1)^2 - 4R^2}}{2R}, \qquad p = \frac{R-d}{u-d}. \tag{6.1.2}$$

The expression for u in the above formula appears to be quite cumbersome. It is tempting to seek a simpler formula for u, while not sacrificing the order of accuracy. By expanding u in Taylor series in powers of $\sqrt{\triangle t}$, we obtain

$$u = 1 + \sigma\sqrt{\triangle t} + \frac{\sigma^2}{2}\triangle t + \frac{4r^2 + 4\sigma^2 r + 3\sigma^4}{8\sigma}\triangle t^{\frac{3}{2}} + O(\triangle t^2). \tag{6.1.3}$$

Observe that the first three terms in the above Taylor series agree with those of $e^{\sigma\sqrt{\triangle t}}$ up to $O(\triangle t)$ term. This suggests the judicious choice of the following set of parameter values (Cox *et al.*, 1979; Chap. 2)

$$u = e^{\sigma\sqrt{\triangle t}}, \quad d = e^{-\sigma\sqrt{\triangle t}}, \qquad p = \frac{R-d}{u-d}. \tag{6.1.4}$$

This set of parameters appear to be simpler compared to those in formula (6.1.2). With this new set of parameters, the variance of the price ratio $\frac{S_{t+\triangle t}}{S_t}$ in the continuous and discrete models agree up to $O(\triangle t)^2$. More precisely, Eq. (6.1.1b) is now satisfied up to $O(\triangle t^2)$ since

$$pu^2 + (1-p)d^2 - R^2 e^{\sigma^2 \triangle t} = -\frac{5\sigma^4 + 12r\sigma^2 + 12r^2}{12}\triangle t^2 + O(\triangle t^3). \tag{6.1.5}$$

Other choices of parameter values have been proposed in the literature (see Problem 6.1). They all share the same order of accuracy in approximating Eq. (6.1.1b), but their analytic expressions are more cumbersome. This explains why the parameter values in Eq (6.1.4) are most commonly used in binomial models.

6.1.2 Continuous limits of the binomial model

Given the parameter values for u, d and p in Eq. (6.1.4), we consider the asymptotic limit $\triangle t \to 0$ of the binomial formula

$$c = [pc_u^{\Delta t} + (1-p)c_d^{\Delta t}]\, e^{-r\triangle t}, \tag{6.1.6}$$

We would like to show that the Black-Scholes equation for the continuous model is obtained as a result. First, it is necessary to perform continuation of the grid function to continuous function such that the two functions agree with each other at the node points. In the continuous analog, the binomial formula can be written as

$$c(S, t - \triangle t) = [pc(uS, t) + (1-p)c(dS, t)]\, e^{-r\triangle t}. \tag{6.1.7}$$

Here, for the convenience of presentation, we take the current time to be $t - \Delta t$. Assuming sufficient continuity of $c(S,t)$, we perform the Taylor expansion of the binomial scheme at (S,t) as follows:

$$\begin{aligned}
&- c(S, t-\triangle t) + [pc(uS,t) + (1-p)c(dS,t)]e^{-r\triangle t} \\
&= \frac{\partial c}{\partial t}(S,t)\triangle t - \frac{1}{2}\frac{\partial^2 c}{\partial t^2}(S,t)\triangle t^2 + \cdots - (1 - e^{-r\triangle t})c(S,t) \\
&\quad + e^{-r\triangle t}\left\{[p(u-1) + (1-p)(d-1)]S\frac{\partial c}{\partial S}(S,t)\right. \\
&\quad + \frac{1}{2}[p(u-1)^2 + (1-p)(d-1)^2]S^2\frac{\partial^2 c}{\partial S^2}(S,t) \\
&\quad \left. + \frac{1}{6}[p(u-1)^3 + (1-p)(d-1)^3]S^3\frac{\partial^3 c}{\partial S^3}(S,t) + \cdots\right\}.
\end{aligned} \tag{6.1.8}$$

By observing that

$$1 - e^{-r\triangle t} = r\triangle t + O(\triangle t^2), \tag{6.1.9a}$$

it can be shown that

$$e^{-r\triangle t}\,[p(u-1) + (1-p)(d-1)] = r\triangle t + O(\triangle t^2), \tag{6.1.9b}$$

$$e^{-r\triangle t}\,[p(u-1)^2 + (1-p)(d-1)^2] = \sigma^2\triangle t + O(\triangle t^2), \tag{6.1.9c}$$

$$e^{-r\triangle t}\,[p(u-1)^3 + (1-p)(d-1)^3] = O(\triangle t^2). \tag{6.1.9d}$$

Substituting the above results into Eq. (6.1.8), we obtain

$$\begin{aligned}
&- c(S, t-\triangle t) + [pc(uS,t) + (1-p)c(dS,t)]\, e^{-r\triangle t} \\
&= \left[\frac{\partial c}{\partial t}(S,t) + rS\frac{\partial c}{\partial S}(S,t) + \frac{\sigma^2}{2}S^2\,\frac{\partial^2 c}{\partial S^2}(S,t) - rc(S,t)\right]\triangle t + O(\triangle t^2).
\end{aligned} \tag{6.1.10}$$

Since $c(S,t)$ satisfies the binomial formula (6.1.7), so we obtain

$$0 = \frac{\partial c}{\partial t}(S,t) + rS\frac{\partial c}{\partial S}(S,t) + \frac{\sigma^2}{2}S^2\frac{\partial^2 c}{\partial S^2}(S,t) - rc(S,t) + O(\triangle t). \tag{6.1.11}$$

In the limit $\Delta t \to 0$, the binomial call value $c(S,t)$ satisfies the Black-Scholes equation. More precisely, the binomial formula approximates the Black-Scholes equation to first order accuracy in time.

Asymptotic limit to the Black-Scholes price formula
We have seen that the continuous limit of the binomial formula tends to the Black-Scholes equation. One would expect that the call price formula for the n-period binomial model [see Eq. (2.2.35)] also tends to the Black-Scholes call price formula in the limit $n \to \infty$, or equivalently $\triangle t \to 0$ (since $n\triangle t$ is finite). Mathematically, we would like to show

$$\lim_{n\to\infty}[S\Phi(n,k,p') - XR^{-n}\Phi(n,k,p)] = SN(d_1) - Xe^{-r\tau}N(d_2), \quad (6.1.12)$$

where

$$d_1 = \frac{\ln\frac{S}{X} + \left(r + \frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau}. \quad (6.1.13)$$

The proof of the above asymptotic result relies on the following well-known result about normal approximation to binomial distribution. Let Y be a binomial random variable with parameters n and p, where n is the number of binomial trials and p is the probability of success. For large n, Y is approximately normal with mean np and variance $np(1-p)$.

To prove formula (6.1.12), it suffices to show

$$\lim_{n\to\infty}\Phi(n,k,p) = N\left(\frac{\ln\frac{S}{X} + (r - \frac{\sigma^2}{2})\tau}{\sigma\sqrt{\tau}}\right) \quad (6.1.14a)$$

and

$$\lim_{n\to\infty}\Phi(n,k,p') = N\left(\frac{\ln\frac{S}{X} + (r + \frac{\sigma^2}{2})\tau}{\sigma\sqrt{\tau}}\right), \quad \tau = T - t. \quad (6.1.14b)$$

The proof of Eq. (6.1.14a) will be presented below while that of Eq. (6.1.14b) is relegated to Problem 6.3.

Recall that $\Phi(n,k,p)$ is the probability that the number of upward moves in the asset price is greater than or equal to k in the n-period binomial model, where p is the probability of an upward move. Let j denote the random integer variable that gives the number of upward moves during the n periods. Consider

$$1 - \Phi(n,k,p) = P[j < k-1] = P\left[\frac{j - np}{\sqrt{np(1-p)}} < \frac{k-1-np}{\sqrt{np(1-p)}}\right], \quad (6.1.15)$$

where $\frac{j - np}{\sqrt{np(1-p)}}$ is the normalized binomial variable. Let S and S^* denote the asset price at the current time and n periods later, respectively. Since S and S^* are related by $S^* = u^j d^{n-j} S$, we then have

$$\ln\frac{S^*}{S} = j\ln\frac{u}{d} + n\ln d. \quad (6.1.16)$$

For the binomial random variable j, its mean and variance are known to be $E(j) = np$ and $\text{var}(j) = np(1-p)$, respectively. Since $\ln \dfrac{S^*}{S}$ and j are linearly related, the mean and variance of $\ln \dfrac{S^*}{S}$ are given by

$$E\left[\ln \frac{S^*}{S}\right] = E[j] \ln \frac{u}{d} + n \ln d = n\left(p \ln \frac{u}{d} + \ln d\right) \tag{6.1.17a}$$

$$\text{var}\left(\ln \frac{S^*}{S}\right) = \text{var}(j)\left(\ln \frac{u}{d}\right)^2 = np(1-p)\left(\ln \frac{u}{d}\right)^2. \tag{6.1.17b}$$

In the limit $n \to \infty$, the mean and variance of the logarithm of the price ratio of the discrete binomial model and the continuous Black-Scholes model should agree with each other, that is,

$$\lim_{n\to\infty} n\left(p \ln \frac{u}{d} + \ln d\right) = \left(r - \frac{\sigma^2}{2}\right)(T-t) \tag{6.1.18a}$$

$$\lim_{n\to\infty} np(1-p)\left(\ln \frac{u}{d}\right)^2 = \sigma^2(T-t), \quad T = t + n\Delta t. \qquad (6.1.18b).$$

Since k is the smallest non-negative integer greater than or equal to $\dfrac{\ln \frac{X}{Sd^n}}{\ln \frac{u}{d}}$, we have

$$k - 1 = \frac{\ln \frac{X}{Sd^n}}{\ln \frac{u}{d}} - \alpha, \qquad \text{where} \quad 0 < \alpha \leq 1, \tag{6.1.19}$$

so that Eq. (6.1.15) can be rewritten as

$$\begin{aligned} 1 - \Phi(n,k,p) &= P[j < k-1] \\ &= P\left[\frac{j - np}{\sqrt{np(1-p)}} < \frac{\ln \frac{X}{S} - n(p \ln \frac{u}{d} + \ln d) - \alpha \ln \frac{u}{d}}{\sqrt{np(1-p)} \ln \frac{u}{d}}\right]. \end{aligned} \tag{6.1.20}$$

In the limit $n \to \infty$, or equivalently $\Delta t \to 0$, the quantities $\sqrt{np(1-p)} \ln \dfrac{u}{d}$ and $n\left(p \ln \dfrac{u}{d} + \ln d\right)$ are finite [see Eqs. (6.1.18a,b)] while $\alpha \ln \dfrac{u}{d}$ is $O(\sqrt{\Delta t})$. By virtue of the property of normal approximation to the binomial distribution and the asymptotic results in Eqs. (6.1.18a,b), we obtain

$$\lim_{n\to\infty} \Phi(n,k,p) = 1 - N\left(\frac{\ln \frac{X}{S} - (r - \frac{\sigma^2}{2})\tau}{\sigma\sqrt{\tau}}\right) = N\left(\frac{\ln \frac{S}{X} + \left(r - \frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\right), \tag{6.1.21}$$

where $\tau = T - t$.

6.1.3 Discrete dividend models

The binomial model can easily incorporate the effect of dividend yield paid by the underlying asset (see Problem 6.2). With some simplifying but reasonable assumptions, we can also incorporate discrete dividends into the discrete binomial model quite effectively.

First, we consider the naive construction of the binomial tree. Let S be the asset price at the current time which is $n\triangle t$ from expiry, and suppose a discrete dividend of amount D is paid at time between one time step and two time steps from the current time. The nodes in the binomial tree at two time steps from the current time would correspond to asset prices

$$u^2S - D, \quad S - D \quad \text{and} \quad d^2S - D,$$

since the asset price drops by the same amount as the dividend right after the dividend payment (see Fig. 6.1). Extending one time step further, there will be six nodes

$$(u^2S - D)u, (u^2S - D)d, (S - D)u, (S - D)d, (d^2S - D)u, (d^2S - D)d$$

instead of four nodes as in the usual binomial tree without discrete dividend. This is because $(u^2S - D)d \neq (S - D)u$ and $(S - D)d \neq (d^2S - D)u$, so the interior nodes do not recombine. Extending one time step further, the number of nodes will grow to nine instead of five as in the usual binomial tree. In general, suppose a discrete dividend is paid in the future between k and $k + 1$ time steps from the current time, then at $k + m$ time steps later from the current time, the number of nodes would be $m(k + 1)$ rather than $k + m + 1$ as in the usual reconnecting binomial tree.

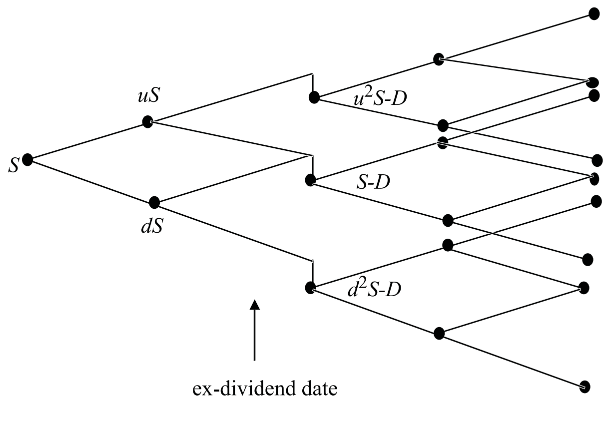

Fig. 6.1 Binomial tree with single discrete dividend.

The above difficulty of nodes exploding can be circumvented by splitting the asset price S_t into two parts: the risky component $\widetilde{S}_t$ that is stochastic and the remaining part that will be used to pay the discrete dividend (assumed to be deterministic) in the future. Suppose the dividend date is t^*, then at the current time t, the risky component $\widetilde{S}_t$ is given by [see Eq. (3.4.17)]

$$\widetilde{S}_t = \begin{cases} S_t - De^{-r(t^*-t)}, & t < t^* \\ S_t, & t > t^*. \end{cases} \tag{6.1.22}$$

Let $\widetilde{\sigma}$ denote the volatility of $\widetilde{S}_t$ and assume $\widetilde{\sigma}$ to be constant rather than the volatility of S_t itself to be constant. Now, $\widetilde{\sigma}$ will be used instead of σ in the calculation of the binomial parameters: p, u and d, and a binomial tree is built to model the jump process for $\widetilde{S}_t$. Such assumption is similar in spirit as the common practice of using the Black-Scholes price formula with the asset price reduced by the present value of the dividends. Now, the nodes in the tree for $\widetilde{S}_t$ become reconnected and adding the present value of the dividend at nodes before the dividend date will give the reconnecting tree for S_t.

Let S and $\widetilde{S}$ denote the asset price and its risky component at the tip of the binomial tree for S_t, respectively, and let N denote the total number of time steps in the tree. Assume that a discrete dividend D is paid at time t^*, which lies between the k^{th} and $(k+1)^{\text{th}}$ time step. At the tip of the binomial tree, the risky component $\widetilde{S}$ is related to the asset price S by

$$\widetilde{S} = S - De^{-kr\Delta t}. \tag{6.1.23}$$

The risky component of the asset price at the $(n, j)^{\text{th}}$ node, which corresponds to n time steps from the tip and j upward jumps, is given by

$$Su^j d^{n-j} - De^{-(k-n)r\Delta t}\mathbf{1}_{\{n \le k\}},$$
$$n = 1, 2, \cdots, N \quad \text{and} \quad j = 0, 1, \cdots, n.$$

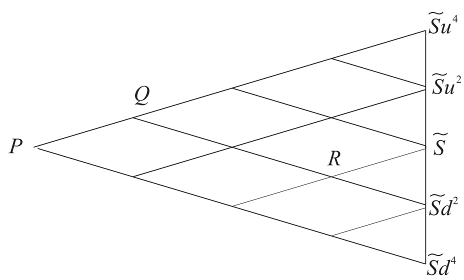

Fig. 6.2 Construction of a reconnecting binomial tree with single discrete dividend D. Here, $N = 4$ and $k = 2$, and let $\widetilde{S}$ denote the risky component of the asset value at the tip of the bonomial tree. The asset value at nodes P, Q and R are $\widetilde{S}+De^{-2r\Delta t}$, $\widetilde{S}\,u+De^{-r\Delta t}$ and $\widetilde{S}d$, respectively.

Once the reconnecting tree for S is available, the option values at the nodes can be found using the binomial formula using backward induction. It is quite straightforward to generalize the above splitting approach to option models with several discrete dividends.

6.1.4 Early exercise feature and callable feature

Recall that an American option can be terminated prematurely due to possibility of early exercise by the holder. Without the early exercise privilege, risk neutral valuation leads to the usual binomial formula

$$V_{cont} = \frac{pV_u^{\Delta t} + (1-p)V_d^{\Delta t}}{R}. \tag{6.1.24}$$

Here, we use V_{cont} to represent the state of continuation value where the option is kept alive. To incorporate the early exercise possibility embedded in an American option, we compare at each binomial node the continuation value V_{cont} with the option's intrinsic value, which is the payoff function upon exercise. The following simple dynamic programming procedure is applied at each binomial node

$$V = \max(V_{cont}, h(S)), \tag{6.1.25}$$

where $h(S)$ is the exercise payoff when the asset price assumes the value S.

As an example, we consider the valuation of an American vanilla put option. First, we build the usual binomial tree which gives a discrete representation of the stochastic movement of the asset price (with or without dividend). Here, N denotes the number of time steps from the current time to expiry. Let S_j^n and P_j^n denote the asset price and put value at the $(n,j)^{\text{th}}$ node, respectively. The intrinsic value of a vanilla put option is $X - S_j^n$ at the (n,j) node, where X is the strike price. Hence, the dynamic programming procedure applied at each node is given by

$$P_j^n = \max\left(\frac{pP_{j+1}^{n+1} + (1-p)P_j^{n+1}}{R}, X - S_j^n\right), \tag{6.1.26}$$

where $n = N-1, \cdots 0$, and $j = 0, 1, \cdots, n$.

Many enhanced numerical schemes for valuation of American options have been proposed in the literature (Dempster and Hutton, 1999). A good survey of comparison of their numerical performance can be found in Broadie and Detemple's paper (1996).

Also, the binomial scheme can be easily modified to incorporate additional embedded features in an American option contract. For example, the callable feature entitles the issuer to buy back the American option at any time at a predetermined call price. Upon call, the holder can choose either to exercise the call or receive the call price as cash. Consider a callable American call option with call price K. To price such call, the dynamic programming procedure applied at each node is modified as follows (Kwok and Wu, 2000; Chap. 5)

$$C_j^n = \min\left(\max\left(\frac{pc_{j+1}^{n+1} + (1-p)c_j^{n+1}}{R}, S_j^n - X\right),\right.$$
$$\left.\max(K, S_j^n - X)\right). \tag{6.1.27}$$

The first term $\max\left(\frac{pc_{n+1}^{n+1} + (1-p)c_j^{n+1}}{R}, S_j^n - X\right)$ represents the optimal strategy of the holder, given no call of the option by the issuer. Upon call by the issuer, the payoff is given by the second term $\max(K, S_j^n - X)$ since the holder can either receive cash amount K or exercise the option. From the perspective of the issuer, he chooses to call or restrain from calling so as to minimize the option value with reference to the possible actions of the holder. Hence, the value of the callable call is given by taking the minimum value of the above two terms.

There are several other alternative forms of the binomial schemes to price the callable American call option. For details, see Problems 6.6 and 6.7.

6.1.5 Trinomial schemes

In binomial models, we assume a two-jump process for the asset price over each discrete time step. One may query whether accuracy and reliability of option valuation can be improved by allowing a three-jump process for the stochastic asset price. In a trinomial model, the asset price S is assumed to jump to either uS, mS or dS after one time period $\triangle t$, where $u > m > d$. We consider a trinomial formula of option valuation of the form

$$V = \frac{p_1 V_u^{\Delta t} + p_2 V_m^{\Delta t} + p_3 V_d^{\Delta t}}{R}, \qquad R = e^{r\triangle t}. \tag{6.1.28}$$

Here, $V_u^{\Delta t}$ denotes the option price when the asset price takes the value uS one period later, and similar interpretation for $V_m^{\Delta t}$ and $V_d^{\Delta t}$. The new trinomial model may allow greater freedom in the selection of parameters to achieve some desirable properties, like avoiding negative probabilities, attaining a faster rate of convergence, etc. The tradeoff is lowering of computational efficiency in general since a trinomial scheme requires more computational steps compared to that of a binomial scheme (see Problem 6.8). Cox *et al.* (1979) caution that the trinomial model (unlike the binomial model) will not lead to an option pricing formula based solely on arbitrage considerations. However, a direct link between the approximating process of the asset price movement and the arbitrage strategy is not essential. In fact, any contingent claim can be valued by computing conditional expectation under an appropriate measure. If such conditional expectation is difficult to evaluate, one may use an approximating discrete process to approximate the underlying asset price movement. The different approximating procedures lead to different numerical schemes.

Recall that under the risk neutral measure, $\ln \dfrac{S_{t+\triangle t}}{S_t}$ is normally distributed with mean $\left(r - \dfrac{\sigma^2}{2}\right)\triangle t$ and variance $\sigma^2 \triangle t$. Alternatively, we may write

$$\ln S_{t+\triangle t} = \ln S_t + \zeta, \tag{6.1.29}$$

where ζ is a normal random variable with mean $\left(r - \dfrac{\sigma^2}{2}\right)\triangle t$ and variance $\sigma^2 \triangle t$. Kamrad and Ritchken (1991) propose to approximate ζ by an approximate discrete random variable ζ^a with the following distribution

$$\zeta^a = \begin{cases} v & \text{with probability } p_1 \\ 0 & \text{with probability } p_2 \\ -v & \text{with probability } p_3 \end{cases} \tag{6.1.30}$$

where $v = \lambda\sigma\sqrt{\triangle t}$ and $\lambda \geq 1$. The corresponding values for u, m and d in the trinomial scheme are: $u = e^v, m = 1$ and $d = e^{-v}$. To find the probability values p_1, p_2 and p_3, the mean and variance of ζ^a are chosen to be equal to those of ζ. These lead to

$$E[\zeta^a] = v(p_1 - p_3) = \left(r - \frac{\sigma^2}{2}\right)\triangle t \tag{6.1.31a}$$

$$\text{var}(\zeta^a) = v^2(p_1 + p_3) - v^2(p_1 - p_3)^2 = \sigma^2 \triangle t. \tag{6.1.31b}$$

From Eq. (6.1.31a), we see that $v^2(p_1 - p_3)^2 = O(\Delta t^2)$. We may drop this term from Eq. (6.1.31b) so that

$$v^2(p_1 + p_3) = \sigma^2 \triangle t, \tag{6.1.31c}$$

while still maintaining $O(\Delta t)$ accuracy. Without this simplication, the final expressions for p_1, p_2 and p_3 would become more cumbersome. Lastly, the probabilities must be summed to one so that

$$p_1 + p_2 + p_3 = 1. \tag{6.1.32}$$

We then solve Eqs. (6.1.31a,c) and (6.1.32) together to obtain

$$p_1 = \frac{1}{2\lambda^2} + \frac{(r - \frac{\sigma^2}{2})\sqrt{\triangle t}}{2\lambda\sigma} \tag{6.1.33a}$$

$$p_2 = 1 - \frac{1}{\lambda^2} \tag{6.1.33b}$$

$$p_3 = \frac{1}{2\lambda^2} - \frac{(r - \frac{\sigma^2}{2})\sqrt{\triangle t}}{2\lambda\sigma}. \tag{6.1.33c}$$

The expressions for the probabilities appear to be much simpler than that of Boyle's trinomial model (see Problem 6.10). By choosing different values for the free parameter λ, a range of probability values can be obtained. In particular, when $\lambda = 1$, we obtain $p_2 = 0$. In this case, the trinomial scheme reduces to a binomial scheme.

Numerical experiments have revealed that when λ is chosen such that the horizontal jump probability is about one-third, the errors in the approximation are minimized. Though a trinomial scheme is seen to require more computational work than that of a binomial scheme, one can show easily that a trinomial scheme with n steps requires less computational work (measured in terms of number of multiplications and additions) than a binomial scheme with $2n$ steps (see Problem 6.8). The numerical tests performed by Kamrad and Ritchken (1991) reveal that the trinomial scheme with n steps invariably performs better in accuracy than the binomial scheme with $2n$ steps. In terms of order of accuracy, both the binomial scheme and trinomial scheme satisfy the Black-Scholes equation to first-order accuracy (see Problem 6.11).

Multi-state options

The extension of the above approach to two-state options is quite straightforward. First, we assume the joint density of the prices of the two underlying assets S_1 and S_2 to be bivariate lognormal. Let σ_i be the volatility of asset price S_i, $i = 1, 2$ and ρ be the correlation coefficient between the two lognormal diffusion processes. Let S_i and $S_i^{\triangle t}$ denote, respectively, the price of asset i at the current time and one period $\triangle t$ later. Under the risk neutral measure, we have

$$\ln \frac{S_i^{\triangle t}}{S_i} = \zeta_i, \qquad i = 1, 2, \tag{6.1.34}$$

where ζ_i is a normal random variable with mean $\left(r - \frac{\sigma_i^2}{2}\right)\triangle t$ and variance $\sigma_i^2 \triangle t$. The instantaneous correlation coefficient between ζ_1 and ζ_2 is ρ. The joint bivariate normal process $\{\zeta_1, \zeta_2\}$ is approximated by a pair of joint discrete random variables $\{\zeta_1^a, \zeta_2^a\}$ with the following distribution

ζ_1^a	ζ_2^a	probability
v_1	v_2	p_1
v_1	$-v_2$	p_2
$-v_1$	$-v_2$	p_3
$-v_1$	v_2	p_4
0	0	p_5

where $v_i = \lambda_i \sigma_i \sqrt{\triangle t}$, $i = 1, 2$. There are five probability values to be determined. In our approximation procedures, we set the first two moments of the approximating distribution (including the covariance) to the corresponding moments of the continuous distribution. Equating the corresponding means gives

$$E[\zeta_1^a] = v_1(p_1 + p_2 - p_3 - p_4) = \left(r - \frac{\sigma_1^2}{2}\right) \triangle t \tag{6.1.35a}$$

$$E[\zeta_2^a] = v_2(p_1 - p_2 - p_3 + p_4) = \left(r - \frac{\sigma_2^2}{2}\right) \triangle t. \tag{6.1.35b}$$

By equating the variances and covariance to $O(\triangle t)$ accuracy, we have

$$\text{var}(\zeta_1^a) = v_1^2(p_1 + p_2 + p_3 + p_4) = \sigma_1^2 \triangle t \tag{6.1.35c}$$

$$\text{var}(\zeta_2^a) = v_2^2(p_1 + p_2 + p_3 + p_4) = \sigma_2^2 \triangle t \tag{6.1.35d}$$

$$E[\zeta_1^a \zeta_2^a] = v_1 v_2(p_1 - p_2 + p_3 - p_4) = \sigma_1 \sigma_2 \rho \triangle t. \tag{6.1.35e}$$

In order that Eqs. (6.1.35c,d) are consistent, we must set $\lambda_1 = \lambda_2$. Writing $\lambda = \lambda_1 = \lambda_2$, we have the following four independent equations for the five probability values

$$p_1 + p_2 - p_3 - p_4 = \frac{(r - \frac{\sigma_1^2}{2})\sqrt{\triangle t}}{\lambda \sigma_1} \tag{6.1.36a}$$

$$p_1 - p_2 - p_3 + p_4 = \frac{(r - \frac{\sigma_2^2}{2})\sqrt{\triangle t}}{\lambda \sigma_2} \tag{6.1.36b}$$

$$p_1 + p_2 + p_3 + p_4 = \frac{1}{\lambda^2} \tag{6.1.36c}$$

$$p_1 - p_2 + p_3 - p_4 = \frac{\rho}{\lambda^2}. \tag{6.1.36d}$$

Since the probabilities must be summed to one, this gives the remaining condition as

$$p_1 + p_2 + p_3 + p_4 + p_5 = 1. \tag{6.1.36e}$$

The solution of the above linear algebraic system of equations gives

$$p_1 = \frac{1}{4}\left[\frac{1}{\lambda^2} + \frac{\sqrt{\triangle t}}{\lambda}\left(\frac{r - \frac{\sigma_1^2}{2}}{\sigma_1} + \frac{r - \frac{\sigma_2^2}{2}}{\sigma_2}\right) + \frac{\rho}{\lambda^2}\right] \tag{6.1.37a}$$

$$p_2 = \frac{1}{4}\left[\frac{1}{\lambda^2} + \frac{\sqrt{\triangle t}}{\lambda}\left(\frac{r - \frac{\sigma_1^2}{2}}{\sigma_1} - \frac{r - \frac{\sigma_2^2}{2}}{\sigma_2}\right) - \frac{\rho}{\lambda^2}\right] \tag{6.1.37b}$$

$$p_3 = \frac{1}{4}\left[\frac{1}{\lambda^2} + \frac{\sqrt{\triangle t}}{\lambda}\left(-\frac{r - \frac{\sigma_1^2}{2}}{\sigma_1} - \frac{r - \frac{\sigma_2^2}{2}}{\sigma_2}\right) + \frac{\rho}{\lambda^2}\right] \tag{6.1.37c}$$

$$p_4 = \frac{1}{4}\left[\frac{1}{\lambda^2} + \frac{\sqrt{\triangle t}}{\lambda}\left(-\frac{r - \frac{\sigma_1^2}{2}}{\sigma_1} + \frac{r - \frac{\sigma_2^2}{2}}{\sigma_2}\right) - \frac{\rho}{\lambda^2}\right] \tag{6.1.37d}$$

$$p_5 = 1 - \frac{1}{\lambda^2}, \qquad \lambda \geq 1 \text{ is a free parameter.} \tag{6.1.37e}$$

For convenience, we write $u_i = e^{v_i}$, $d_i = e^{-v_i}$, $i = 1, 2$. Let V denote the price of a two-state option with underlying asset prices S_1 and S_2. Also, let $V^{\Delta t}_{u_1 u_2}$ denote the option price at one time period later with asset prices $u_1 S_1$ and $u_2 S_2$, and similar meaning for $V^{\Delta t}_{u_1 d_2}, V^{\Delta t}_{d_1 u_2}$ and $V^{\Delta t}_{d_1 d_2}$. We let $V^{\Delta t}_{0,0}$ denote the option price one period later with no jumps in asset prices. The corresponding 5-point formula for the two-state trinomial model can be expressed as (Kamrad and Ritchken, 1991)

$$V = (p_1 V^{\triangle t}_{u_1 u_2} + p_2 V^{\triangle t}_{u_1 d_2} + p_3 V^{\triangle t}_{d_1 d_2} + p_4 V^{\triangle t}_{d_1 u_2} + p_5 V^{\triangle t}_{0,0})/R. \qquad (6.1.38)$$

In particular, when $\lambda = 1$, we have $p_5 = 0$ and the above 5-point formula reduces to the 4-point formula.

The presence of the free parameter λ in the 5-point formula provides the flexibility to explore better convergence behavior of the discrete pricing formula. With proper choice of λ, Kamrad and Ritchken (1991) observe from their numerical experiments that convergence of the numerical values obtained from the 5-point formula to the continuous solution is invariably smoother and more rapid than those obtained from the 4-point formula. The extension of the present approach to three-state option models can be derived in a similar manner (see Problem 6.14).

6.1.6 Forward shooting grid methods

For path dependent options, the option value also depends on the path function $F_t = F(S, t)$ defined specifically for the given nature of path dependence. For example, the path dependence may be defined by the minimum asset price realized along a specific asset price path. Since option value depends also on F_t, we find the value of the path dependent option at each node in the lattice tree for all alternative values of F_t that can occur. In order that the numerical scheme competes well in terms of efficiency, it is desirable that the value $F_{t+\Delta t}$ can be computed easily from F_t and $S_{t+\triangle t}$ (that is, the path function is Markovian) and the number of alternative values for $F(S, t)$ cannot grow too large with increasing number of binomial steps. The approach of appending an auxiliary state vector at each node in the lattice tree to model the correlated evolution of F_t with S_t is commonly called the *forward shooting grid* (*FSG*) *method*.

The FSG approach is pioneered by Hull and White (1993) for pricing American and European Asian and lookback options. A systematic framework of constructing FSG schemes for pricing path dependent options is presented by Barraquand and Pudet (1996). Forsyth *et al*. (2002) show that convergence of the numerical solutions of the FSG schemes for pricing Asian options depend on the method of interpolation of the average asset values between neighboring lattice nodes. The methods of interpolation include nearest node interpolation, linear and quadratic interpolation. Jiang and Dai (2004) use the notion of viscosity solution to show uniform convergence of the FSG schemes for pricing American and European arithmetic Asian options.

For some exotic path dependent options, like the window Parisian option (see Problem 6.16), the governing option pricing equation cannot be derived. However, by relating the correlated evolution of the augmented path dependent state variable with the asset price, it is still possible to devise the FSG schemes for pricing these exotic options.

Consider a trinomial tree whose probabilities of upward, zero and downward jump of the asset price are denoted by p_u, p_0 and p_d, respectively. Let $V_{j,k}^n$ denote the numerical option value of the exotic path dependent option at the n^{th}-time level (n time steps from the tip of the tree). Also, j denotes the j upward jumps from the initial asset value and k denotes the numbering index for the various possible values of the augmented state variable F_t at the $(n, j)^{\text{th}}$ node. Let G denote the function that describes the correlated evolution of F_t with S_t over the time interval Δt, that is,

$$F_{t+\Delta t} = G(F_t, S_{t+\Delta t}). \tag{6.1.39}$$

Let $g(k, j)$ denote the grid function which is considered as the discrete analog of the evolution function G. The trinomial version of the FSG scheme can be represented as follows

$$V_{j,k}^n = \left[p_u V_{j+1,g(k,j+1)}^{n+1} + p_0 V_{j,g(k,j)}^{n+1} + p_d V_{j-1,g(k,j-1)}^{n+1} \right] e^{-r\Delta t}, \tag{6.1.40}$$

where $e^{-r\Delta t}$ is the discount factor over time interval Δt. To price a specific path dependent option, the design of the FSG algorithm requires the specification of the grid function $g(k, j)$. We illustrate how to find $g(k, j)$ for various types of path dependent options, which include options with Parisian variant of knock-out feature and Asian options.

Options with Parisian variant of knock-out

The one-touch breaching of barrier in barrier options has the undesirable effect of knocking out the option when the asset price spikes, no matter how briefly the spiking occurs. Hedging barrier options may become difficult when the asset price is very close to the barrier. In the foreign exchange markets, market volatility may increase around popular barrier levels due to plausible price manipulation aimed at activating knock-out.

To circumvent the spiking effect and short-period price manipulation, various forms of Parisian knock-out provision have been proposed in the literature. Here, knock-out is activated only when the underlying asset price breaches the barrier for a prespecified period of time. The breaching can be counted consecutively or cumulatively. In actual market practice, breaching is monitored at discrete time instants rather than continuously, so the number of breaching occurrences at monitoring instants is counted. Here, we derive the FSG scheme for pricing option with cumulative Parisian feature. The construction of FSG schemes for the consecutive Parisian feature and window Parisian feature are relegated to Problems 6.15 and 6.16. The application of the FSG approach to price convertible bonds with Parisian variant of soft call requirement can be found in Lau-Kwok's paper (2004).

Cumulative Parisian feature
Let M denote the prespecified number of cumulative breaching occurrences that is required to activate knock-out, and let k be the integer variable that counts the number of breaching so far. Let B denote the down barrier associated with the knock-out feature. Now, the augmented path dependent state variable at each node is the integer k. The value of k is not changed except at time step which corresponds to a monitoring instant. Let $V_{j,k}^n$ denote the value of the option with the cumulative Parisian feature at the $(n, j)^{\text{th}}$ node in a trinomial tree. Let x_j denote the value of $x = \ln S$ that corresponds to j upward moves in the trinomial tree. When $n\Delta t$ happens to be a monitoring instant, the index k increases its value by 1 if the asset price S falls on or below the barrier B, that is, $x_j \leq \ln B$. To incorporate the cumulative Parisian feature, the appropriate choice of the grid function $g_{cum}(k, j)$ is defined by

$$g_{cum}(k, j) = k + \mathbf{1}_{\{x_j \leq \ln B\}}. \tag{6.1.41}$$

The schematic diagram that illustrates the construction of $g_{cum}(k, j)$ is shown in Fig. 6.3.

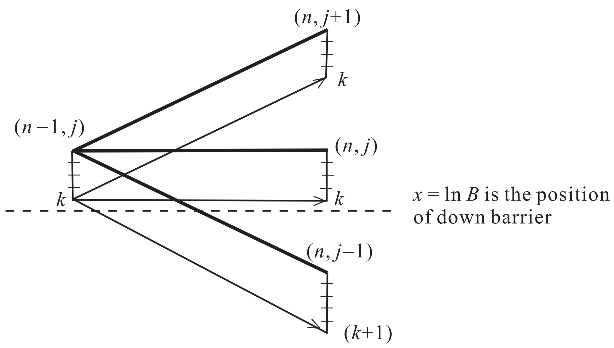

Fig. 6.3 Schematic diagram that illustrates the construction of the grid function $g_{cum}(k, j)$ that models the cumulative Parisian feature. The down barrier $\ln B$ is placed mid-way between two horizontal rows of trinomial nodes. Here, the n^{th}-time level is a monitoring instant.

When $n\Delta t$ is not a monitoring instant, the trinomial tree calculations proceed like those for usual options. Now, the FSG algorithm for pricing an option with the cumulative Parisian feature can be represented by

$$V_{j,k}^{n-1} = \begin{cases} p_u V_{j+1,k}^n + p_0 V_{j,k}^n + p_d V_{j-1,k}^n \\ \qquad \text{if } n\Delta t \text{ is not a monitoring instant} \\ p_u V_{j+1,g_{cum}(k,j+1)}^n + p_0 V_{j,g_{cum}(k,j)}^n + p_d V_{j-1,g_{cum}(k,j-1)}^n \\ \qquad \text{if } n\Delta t \text{ is a monitoring instant} \end{cases}. \quad (6.1.42)$$

In typical FSG calculations, it is necessary to start with $V_{j,M-1}^n$, then $V_{j,M-2}^n, \cdots$, and proceed down until the index k hits 0. We compute $V_{j,M-1}^n$ by setting $k = M-1$ in Eq. (6.1.42) and observe that $V_{j,M}^n = 0$ for all n and j. Actually, $V_{j,M-1}^n$ is the option value of the one-touch down-and-out option at the same node.

Remarks

1. The pricing of options with continuously monitored cumulative Parisian feature is obtained by setting all time steps to be monitoring instant.
2. The computational time required for pricing an option with cumulative Parisian feature requring M breaching occurrences to knock out is about M times that of an one-touch knock-out barrier option.

Floating strike arithmetic averaging Asian call

To price an Asian option, we find the option value at each node for all alternative values of the path function $F(S,t)$ that can occur at that node. Now, the number of possible values for the averaging value F at a binomial node for arithmetic averaging options grows exponentially at 2^n. Therefore, the binomial schemes that place no constraint on the number of possible F values at a node become infeasible for arithmetic averaging options. A possible remedy is to restrict the possible values for F to a certain set of predetermined values. The option value $V(S,F,t)$ for other values of F is obtained from the known values of V at predetermined F values by interpolation (Barraquand and Pudet, 1996; Forsyth *et al.*, 2002).

We illustrate the interpolation technique through valuation of the floating strike arithmetic averaging call option. Here, we define

$$A_t = \frac{1}{t}\int_0^t S_u \, du. \quad (6.1.43)$$

The terminal payoff of the Asian option is given by $\max(S(T) - A(T), 0)$, where A_T is the arithmetic average of S over period $[0,T]$. For a given time step Δt, we fix

$$\Delta W = \sigma\sqrt{\Delta t} \text{ and } \Delta Y = \rho \Delta W, \quad \rho < 1, \quad (6.1.44a)$$

and define the possible values for S_t and A_t at the n^{th} time step by

$$S_j^n = S_0 e^{j\Delta W} \text{ and } A_k^n = S_0 e^{k\Delta Y}, \quad (6.1.44b)$$

where j and k are integers, and S_0 is the asset price at the tip of the binomial tree. By differentiating Eq. (6.1.43) with respect to t, we obtain

$$d(tA_t) = S_t\,dt, \tag{6.1.45a}$$

and from which we deduce the following discrete analog

$$A_{t+\Delta t} = \frac{(t+\Delta t)A_t + \Delta t\ S_{t+\Delta t}}{t+2\Delta t}. \tag{6.1.45b}$$

Consider the binomial procedure at node (n, j), suppose we have an upward move in asset price from S_j^n to S_{j+1}^{n+1} and let $A_{k^+}^{n+1}$ be the corresponding new value of A_t moving from A_k^n. Setting $A_0^0 = S_0$, the equivalence of Eq. (6.1.45b) is given by

$$A_{k^+}^{n+1} = \frac{(n+1)A_k^n + S_{j+1}^{n+1}}{n+2}. \tag{6.1.46a}$$

Similarly, for a downward move in asset price from S_j^n to S_{j-1}^{n+1}, A_k^n changes to $A_{k^-}^{n+1}$ where

$$A_{k^-}^{n+1} = \frac{(n+1)A_k^n + S_{j-1}^{n+1}}{n+2}. \tag{6.1.46b}$$

Note that $A_{k^\pm}^{n+1}$ in general does not coincide with $A_{k'}^{n+1} = Se^{k'\Delta Y}$, for some integer k'. Suppose we define the integers $k_{floor}^{\pm}$ such that $A_{k_{floor}^{\pm}}^{n+1}$ are the largest possible $A_{k'}^{n+1}$ values less than or equal to $A_{k_\pm}^{n+1}$, then the integers k_{floor}^{+} and k_{floor}^{-} are found to be

$$k_{floor}^{\pm} = \text{floor}(k^{\pm}) = \text{floor}\left(\frac{\ln \frac{(n+1)e^{k\Delta Y} + e^{(j\pm 1)\Delta W}}{n+2}}{\Delta Y}\right), \tag{6.1.47}$$

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

What would be the possible range of k at the nth time step? We observe that the average A_t must lie between the maximum asset value S_n^n and the minimum asset value S_{-n}^n, and so k must lie between $-\frac{n}{\rho} \le k \le \frac{n}{\rho}$. Except with very small value for ρ, the number of predetermined values for A_t is in general manageable.

Write $\ell_{floor} = \text{floor}(\ell)$ and let $\ell_{ceil} = \ell_{floor} + 1$, then A_ℓ^n lies between $A_{\ell_{floor}}^n$ and $A_{\ell_{ceil}}^n$. Here, ℓ is a real number in general, while ℓ_{floor} and ℓ_{ceil} are integers. We approximate $c_{j,\ell}^n$ in terms of $c_{j,\ell_{floor}}^n$ and $c_{j,\ell_{ceil}}^n$ by the following linear interpolation formula

$$c_{j,\ell}^n = \epsilon c_{j,\ell_{floor}}^n + (1-\epsilon)c_{j,\ell_{ceil}}^n, \tag{6.1.48a}$$

where

$$\epsilon = \frac{\ln A_\ell^n - \ln A_{\ell_{floor}}^n}{\Delta Y}. \tag{6.1.48b}$$

Following the usual risk neutral valuation approach and applying the above linear interpolation formula (taking ℓ to be k^+ and k^-, successively), the FSG formula for the floating strike arithmetic averaging call option is given by

$$\begin{aligned} c_{j,k}^n &= e^{-r\Delta t}\left[pc_{j+1,k^+}^{n+1} + (1-p)c_{j-1,k^-}^{n+1}\right] \\ &= e^{-r\Delta t}\left\{p\left[\epsilon^+ c_{j+1,k_{ceil}^+}^{n+1} + (1-\epsilon^+)c_{j+1,k_{floor}^+}^{n+1}\right]\right. \\ &\quad \left. + (1-p)\left[\epsilon^- c_{j-1,k_{ceil}^-}^{n+1} + (1-\epsilon^-)c_{j-1,k_{floor}^-}^{n+1}\right]\right\}, \qquad (6.1.49) \end{aligned}$$

$n = N-1, \cdots, 0, j = -n, \cdots, n, k$ is an integer between $-\dfrac{n}{\rho}$ and $\dfrac{n}{\rho}$, $k^\pm$ and $k_{floor}^\pm$ are given by Eq. (6.1.47), and

$$\epsilon^\pm = \frac{\ln A_{k_\pm}^{n+1} - \ln A_{k_{floor}^\pm}^{n+1}}{\Delta Y}. \qquad (6.1.50)$$

The final condition is

$$c_{j,k}^N = S_j^N - A_k^N = S_0 e^{j\Delta W} - S_0 e^{k\Delta Y}, \quad j = -N, \cdots, N, \qquad (6.1.51)$$

k is an integer between $-\dfrac{N}{\rho}$ and $\dfrac{N}{\rho}$.

As a cautious remark, Forsyth *et al.* (2002) prove that the FSG algorithm with nearest lattice point interpolation may exhibit large errors as the number of time steps becomes large. They also show that when linear interpolation is used, the FSG scheme converges to the correct solution plus a constant error term which cannot be reduced by decreasing the size of time step.

6.2 Finite difference algorithms

Finite difference methods are popular numerical techniques for solving science and engineering problems modeled by differential equations. The earliest application of the finite difference methods to option valuation is performed by Brennan and Schwartz (1978). Tavella and Randall's text (2000) contains a comprehensive survey of finite difference methods applied to numerical pricing of financial instruments. In the construction of finite difference schemes, we approximate the differential operators in the governing differential equation of the option model by appropriate finite difference operators, hence the name of this approach.

In this section, we first show how to develop the family of explicit finite difference schemes for option valuation. Interestingly, the binomial and trinomial schemes can be shown to be members in the family of explicit schemes. In explicit schemes, option values at nodes along the new time level can be calculated explicitly from known option values at nodes along the old time level. However, if the discretization of the spatial differential operators involves option values at nodes along the new time level, then the finite difference calculations involve solution of a system of linear equations at every time step. We discuss how implicit finite difference schemes are constructed and the method of their solution using the effective Thomas algorithm. We also consider how to apply finite difference methods for solving American style option models. In the front fixing method, we apply a transformation of variable so that the *front* or free boundary associated with the optimal exercise price is transformed to a *fixed* boundary of the solution domain. Unlike binomial and trinomial schemes, the construction procedure of finite difference scheme allows for direct incorporation of boundary conditions associated with the option models. We illustrate the methods of implementation of the Dirichlet condition in barrier options and Neumann condition in lookback options. To resolve computational nuisance arising from non-differentiability of the "initial" condition, we introduce several effective smoothing techniques that lessen deterioration in accuracy due to non-smooth terminal payoff.

6.2.1 Construction of explicit schemes

Suppose we use the transformed variable: $x = \ln S$, the Black-Scholes equation for the price of a European option becomes

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 V}{\partial x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial V}{\partial x} - rV, \quad -\infty < x < \infty, \tag{6.2.1a}$$

where $V = V(x,\tau)$ is the option value. Here, we adopt time to expiry τ as the temporal variable. Suppose we define $W(x,\tau) = e^{r\tau}U(x,\tau)$, then

$$\frac{\partial W}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial W}{\partial x}, \quad -\infty < x < \infty. \tag{6.2.1b}$$

To derive the finite difference algorithm, we first transform the domain of the continuous problem: $\{(x,\tau) : -\infty < x < \infty, \tau \geq 0\}$ into a discretized domain. The infinite extent of $x = \ln S$ in the continuous problem is approximated by a finite truncated interval $[-M_1, M_2]$, where M_1 and M_2 are sufficiently large positive constants so that the boundary conditions at the two ends of the infinite interval can be applied with sufficient accuracy. The discretized domain is overlaid with a uniform system of meshes or node points $(j\triangle x, n\triangle\tau), j = 0, 1, \cdots, N+1$, where $(N+1)\triangle x = M_1 + M_2$ and $n = 0, 1, 2, \cdots$ (see Fig. 6.4). The stepwidth $\triangle x$ and time step $\triangle\tau$ are in general independent. In the discretized finite difference formulation, the option values are computed only at the node points.

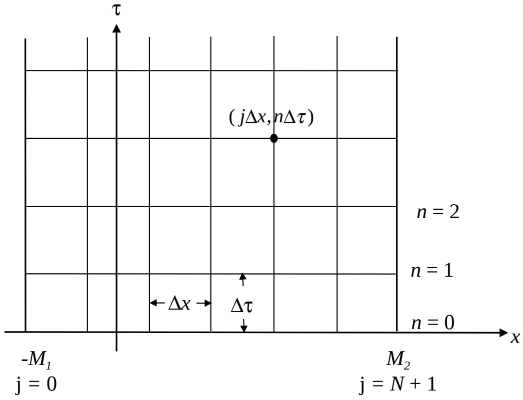

Fig. 6.4 Finite difference mesh with uniform stepwidth Δx and time step $\Delta\tau$. Numerical option values are computed at the node points $(j\Delta x, n\Delta\tau), j = 1, 2, \cdots, N$, $n = 1, 2, \cdots$. Option values along the boundaries: $j = 0$ and $j = N+1$ are prescribed by the boundary conditions of the option model. The "initial" values V_j^0 along the zeroth time level, $n = 0$, are given by the terminal payoff function.

Let V_j^n denote the numerical approximation of $V(j\triangle x, n\triangle\tau)$. The continuous temporal and spatial derivatives in Eq. (6.2.1a) are approximated by the following finite difference operators

$$\frac{\partial V}{\partial \tau}(j\triangle x, n\triangle\tau) \approx \frac{V_j^{n+1} - V_j^n}{\triangle\tau} \qquad \text{(forward difference)} \qquad (6.2.2a)$$

$$\frac{\partial V}{\partial x}(j\triangle x, n\triangle\tau) \approx \frac{V_{j+1}^n - V_{j-1}^n}{2\triangle x} \qquad \text{(centered difference)} \qquad (6.2.2b)$$

$$\frac{\partial^2 V}{\partial x^2}(j\triangle x, n\triangle\tau) \approx \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\triangle x^2} \qquad \text{(centered difference)}. \qquad (6.2.2c)$$

As an intermediate step in the discretization procedure, we also write down the finite difference scheme that discretizes Eq. (6.2.1b) using the above difference approximations. Similarly, we let W_j^n denote the numerical approximation of $W(j\Delta x, n\Delta\tau)$. Next, by observing

$$W_j^{n+1} = e^{r(n+1)\Delta\tau} V_j^{n+1} \quad \text{and} \quad W_j^n = e^{rn\Delta\tau} V_j^n, \qquad (6.2.2d)$$

then canceling $e^{rn\Delta\tau}$, we obtain the following *explicit* Forward-Time-Centered-Space (FTCS) finite difference scheme

$$V_j^{n+1} = \left[V_j^n + \frac{\sigma^2}{2}\frac{\triangle\tau}{\triangle x^2}\left(V_{j+1}^n - 2V_j^n + V_{j-1}^n\right) + \left(r - \frac{\sigma^2}{2}\right)\frac{\triangle\tau}{2\triangle x}\left(V_{j+1}^n - V_{j-1}^n\right)\right]e^{-r\triangle\tau}. \tag{6.2.3}$$

Since V_j^{n+1} is expressed *explicitly* in terms of option values at the n^{th} time level, one can compute V_j^{n+1} directly from known values of V_{j-1}^n, V_j^n and V_{j+1}^n. Suppose we are given "initial" values $V_j^0, j = 0, 1, \cdots, N+1$ along the zeroth time level, we can use scheme (6.2.3) to find values $V_j^1, j = 1, 2, \cdots, N$ along the first time level $\tau = \triangle\tau$. The values at the two ends V_0^1 and V_{N+1}^1 are given by the numerical boundary conditions specified for the option model. In this sense, the boundary conditions are *naturally incorporated* into the finite difference calculations. For example, the Dirichlet boundary conditions in barrier options and Neumann boundary conditions in lookback options can be embedded into the finite difference algorithms (see Sec. 6.2.6 for details). The computational procedure then proceeds in a similar manner to successive time levels $\tau = 2\triangle\tau, 3\triangle\tau, \cdots$, through forward marching along the τ-direction. This is similar to the backward (in the sense of calendar time) valuation in the lattice tree method.

We consider the class of two-level four-point explicit schemes of the form

$$V_j^{n+1} = b_1 V_{j+1}^n + b_0 V_j^n + b_{-1} V_{j-1}^n, \quad j = 1, 2, \cdots, N, \quad n = 0, 1, 2, \cdots \tag{6.2.4}$$

where b_1, b_0 and b_{-1} are coefficients specified for each individual scheme. For example, the above FTCS scheme corresponds to

$$\begin{aligned} b_1 &= \left[\frac{\sigma^2}{2}\frac{\triangle\tau}{\triangle x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\triangle\tau}{2\triangle x}\right]e^{-r\Delta\tau}, \\ b_0 &= \left[1 - \sigma^2\frac{\triangle\tau}{\triangle x^2}\right]e^{-r\Delta\tau}, \\ b_{-1} &= \left[\frac{\sigma^2}{2}\frac{\triangle\tau}{\triangle x^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\triangle\tau}{2\triangle x}\right]e^{r\Delta\tau}. \end{aligned} \tag{6.2.5}$$

An important observation is that both the binomial and trinomial schemes are members of the family specified in Eq. (6.2.4), when the reconnecting condition $ud = 1$ holds. Suppose we write $\triangle x = \ln u$, then $\ln d = -\triangle x$; the binomial scheme can be expressed as

$$V^{n+1}(x) = \frac{pV^n(x + \triangle x) + (1-p)V^n(x - \triangle x)}{R}, \quad x = \ln S, \text{ and } R = e^{r\triangle\tau}, \tag{6.2.6}$$

where $V^{n+1}(x), V^n(x+\triangle x)$ and $V^n(x-\triangle x)$ are analogous to $c, c_u^{\Delta t}$ and $c_d^{\Delta t}$, respectively. The above representation of the binomial scheme corresponds to the following specification of coefficients

$$b_1 = p/R, \quad b_0 = 0 \quad \text{and} \quad b_{-1} = (1-p)/R \tag{6.2.7}$$

in Eq. (6.2.4). Similarly, suppose we choose $\triangle x = \ln u = -\ln d$ and $m = 1$, the trinomial scheme can be expressed as

$$V^{n+1}(x) = \frac{p_1 V^n(x + \triangle x) + p_2 V^n(x) + p_3 V^n(x - \triangle x)}{R}, \tag{6.2.8}$$

which also belongs to the family of explicit schemes defined in Eq. (6.2.4).

While the usual finite difference calculations give option values at all node points along a given time level $\tau = n\triangle\tau$, we compute the option value at single asset value S at $\tau = n\triangle\tau$ in typical binomial/trinomial calculations. For illustration, we consider the computational procedure for the trinomial scheme. Suppose we write $x_j = \ln S$ and n time steps are taken to reach expiry $\tau = 0$ from the current time. The trinomial scheme computes $V^n(x_j)$ from known values of $V^{n-1}(x_{j-1}), V^{n-1}(x_j), V^{n-1}(x_{j+1})$. Down one time level, the computation of $V^n(x_j)$ requires the five values $V^{n-2}(x_{j-2}),\ V^{n-2}(x_{j-1}),\ \cdots, V^{n-2}(x_{j+2})$. Deductively, the $2n + 1$ values $V^0(x_{j-n}),\ V^0(x_{j-n+1}),\ \cdots, V^0(x_{j+n})$ along $\tau = 0$ will be involved to find $V^n(x_j)$. The triangular region in the computational domain with vertices $(x_j, n\triangle\tau)$, $(x_{j-n},\ 0)$ and $(x_{j+n},\ 0)$ is called the *domain of dependence* for the computation of $V^n(x_j)$ (see Fig. 6.5) since the option values at all node points inside the domain of dependence are required for finding $V^n(x_j)$. The practice of confining computation of option values within a triangular domain of dependence is indeed more efficient when only the option value at given S and τ is required.

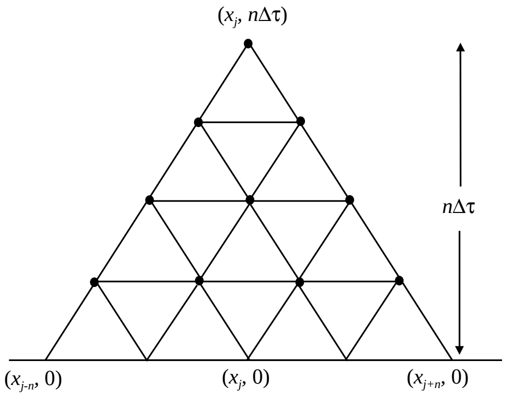

Fig. 6.5 The domain of dependence of a trinomial scheme with n time steps to expiry.

Suppose boundary nodes are not included in the domain of dependence, then the boundary conditions of the option model do not have any effect on the numerical solution of the discrete model. This neglect of boundary conditions does not reduce the accuracy of calculations when the boundary points are at infinity, as in vanilla option models where the domain of definition for $x = \ln S$ is infinite. This is no longer true when the domain of definition for x is truncated, as in barrier option models. To achieve high level of numerical accuracy, it is important that the numerical scheme takes into account the effect of boundary conditions. We will examine the issues of numerical approximation of auxiliary conditions in Sec. 6.2.6.

Note that the stepwidth Δx and time step $\Delta\tau$ in the binomial scheme are dependent. In the Cox-Ross-Rubinstein scheme, they are related by $\Delta x = \ln u = \sigma\sqrt{\Delta\tau}$ or $\sigma^2\Delta\tau = \Delta x^2$. However, in the trinomial scheme, their relation is given by $\lambda^2\sigma^2\Delta\tau = \Delta x^2$, where the free parameter λ can be chosen arbitrarily.

The explicit schemes seem to be easily implementable. However, compared to the implicit schemes discussed in the next subsection, they exhibit lower order of accuracy. Also, the time step in explicit schemes cannot be chosen to be too large due to numerical stability considerations. The concepts of order of accuracy and stability will be explored later in Sec. 6.2.5.

6.2.2 Implicit schemes and their implementation issues

Suppose the discount term $-rV$ and the spatial derivatives are approximated by the average of the centered difference operators at the n^{th} and $(n+1)^{\text{th}}$ time levels

$$\begin{aligned}
-rV\left(j\triangle x, \left(n+\frac{1}{2}\right)\triangle\tau\right) &\approx -\frac{r}{2}\left(V_j^n + V_j^{n+1}\right)\\
\frac{\partial V}{\partial x}\left(j\triangle x, \left(n+\frac{1}{2}\right)\triangle\tau\right) &\approx \frac{1}{2}\left(\frac{V_{j+1}^n - V_{j-1}^n}{2\triangle x} + \frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2\triangle x}\right)\\
\frac{\partial^2 V}{\partial x^2}\left(j\triangle x, \left(n+\frac{1}{2}\right)\triangle\tau\right) &\approx \frac{1}{2}\left(\frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\triangle x^2}\right.\\
&\qquad \left. + \frac{V_{j+1}^{n+1} - 2V_j^{n+1} + V_{j-1}^{n+1}}{\triangle x^2}\right), \qquad (6.2.9a)
\end{aligned}$$

and the temporal derivative by

$$\frac{\partial V}{\partial \tau}\left(j\triangle x, \left(n+\frac{1}{2}\right)\triangle\tau\right) \approx \frac{V_j^{n+1} - V_j^n}{\Delta\tau}, \qquad (6.2.9b)$$

we then obtain the following two-level implicit finite difference scheme

$$
\begin{aligned}
V_j^{n+1} = V_j^n &+ \frac{\sigma^2}{2}\frac{\triangle\tau}{\triangle x^2}\left(\frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n + V_{j+1}^{n+1} - 2V_j^{n+1} + V_{j-1}^{n+1}}{2}\right) \\
&+ \left(r - \frac{\sigma^2}{2}\right)\frac{\triangle\tau}{2\triangle x}\left(\frac{V_{j+1}^n - V_{j-1}^n + V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2}\right) \\
&- r\triangle\tau\left(\frac{V_j^n + V_j^{n+1}}{2}\right),
\end{aligned} \tag{6.2.10}
$$

which is commonly known as the *Crank-Nicolson scheme*.

The above Crank-Nicolson scheme is seen to be a member of the general class of two-level six-point schemes of the form

$$
\begin{aligned}
a_1 V_{j+1}^{n+1} + a_0 V_j^{n+1} + a_{-1} V_{j-1}^{n+1} &= b_1 V_{j+1}^n + b_0 V_j^n + b_{-1} V_{j-1}^n, \\
j = 1, 2, \cdots, N, &\quad n = 0, 1, \cdots
\end{aligned} \tag{6.2.11}
$$

One can observe easily that the Crank-Nicolson scheme corresponds to

$$
\begin{aligned}
a_1 &= -\frac{\sigma^2}{4}\frac{\Delta\tau}{\Delta x^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}, \\
a_0 &= 1 + \frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta x^2} + \frac{r}{2}\Delta\tau, \\
a_{-1} &= -\frac{\sigma^2}{4}\frac{\Delta\tau}{\Delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x},
\end{aligned} \tag{6.2.12a}
$$

and

$$
\begin{aligned}
b_1 &= \frac{\sigma^2}{4}\frac{\Delta\tau}{\Delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}, \\
b_0 &= 1 - \frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta x^2} - \frac{r}{2}\Delta\tau, \\
b_{-1} &= \frac{\sigma^2}{4}\frac{\Delta\tau}{\Delta x^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}.
\end{aligned} \tag{6.2.12b}
$$

A wide variety of implicit finite difference schemes of the class depicted in Eq. (6.2.11) can be derived in a systematic manner (Kwok and Lau, 2001b).

Suppose values for V_j^n are all known along the n^{th} time level, the solution for V_j^{n+1} requires the inversion of a tridiagonal system of equations. This explains the use of the term *implicit* for this class of schemes. In matrix form, the two-level six-point scheme can be represented as

$$
\begin{pmatrix}
a_0 & a_1 & 0 & \cdots & \cdots & 0 \\
a_{-1} & a_0 & a_1 & 0 & \cdots & 0 \\
 & \cdots & & & & \\
 & & \cdots & & & \\
 & & & \cdots & & \\
0 & \cdots & \cdots & 0 & a_{-1} & a_0
\end{pmatrix}
\begin{pmatrix}
V_1^{n+1} \\ V_2^{n+1} \\ \cdot \\ \cdot \\ \cdot \\ V_N^{n+1}
\end{pmatrix}
=
\begin{pmatrix}
c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_N
\end{pmatrix}, \tag{6.2.13}
$$

where

$$\begin{aligned} c_1 &= b_1 V_2^n + b_0 V_1^n + b_{-1} V_0^n - a_{-1} V_0^{n+1}, \\ c_N &= b_1 V_{N+1}^n + b_0 V_N^n + b_{-1} V_{N-1}^n - a_1 V_{N+1}^{n+1}, \\ c_j &= b_1 V_{j+1}^n + b_0 V_j^n + b_{-1} V_{j-1}^n, \quad j = 2, \cdots, N-1. \end{aligned} \tag{6.2.14}$$

The solution of the above tridiagonal system can be effected by the well known *Thomas algorithm*. The algorithm is an efficient implementation of the Gaussian elimination procedure, the details of which are outlined as follows.

Thomas algorithm

Consider the solution of the following tridiagonal system of the form

$$-a_j V_{j-1} + b_j V_j - c_j V_{j+1} = d_j, \quad j = 1, 2, \cdots N, \tag{6.2.15}$$

with $V_0 = V_{N+1} = 0$. This form is more general in the sense that the coefficients can differ among equations. In the first step of elimination, we reduce the system to the upper triangular form by eliminating V_{j-1} in each of the equations. Starting from the first equation, we can express V_1 in terms of V_2 and other known quantities. This relation is then substituted into the second equation giving a new equation involving V_2 and V_3 only. Again, we express V_2 in terms of V_3 and some known quantities. We then substitute into the third equation, ..., and so on.

Suppose the first k equations have been reduced to the form

$$V_j - e_j V_{j+1} = f_j, \quad j = 1, 2, \cdots, k. \tag{6.2.16a}$$

We use the k^{th} reduced equation to transform the original $(k+1)^{\text{th}}$ equation to the same form, namely

$$V_{k+1} - e_{k+1} V_{k+2} = f_{k+1}. \tag{6.2.16b}$$

Now, we consider

$$V_k - e_k V_{k+1} = f_k \tag{6.2.17a}$$

and

$$-a_{k+1} V_k + b_{k+1} V_{k+1} - c_{k+1} V_{k+2} = d_{k+1}, \tag{6.2.17b}$$

the elimination of V_k from these two equations gives a new equation involving V_{k+1} and V_{k+2}, namely,

$$V_{k+1} - \frac{c_{k+1}}{b_{k+1} - a_{k+1} e_k} V_{k+2} = \frac{d_{k+1} + a_{k+1} f_k}{b_{k+1} - a_{k+1} e_k}. \tag{6.2.18}$$

Comparing Eqs. (6.2.16b) and (6.2.18), and replacing the dummy variable $k+1$ by j, we can deduce the following recurrence relations for e_j and f_j:

$$e_j = \frac{c_j}{b_j - a_j e_{j-1}}, \quad f_j = \frac{d_j + a_j f_{j-1}}{b_j - a_j e_{j-1}}, \quad j = 1, 2, \cdots N. \tag{6.2.19}$$

Corresponding to the boundary value $V_0 = 0$, we must have

$$e_0 = f_0 = 0. \tag{6.2.20}$$

Starting from the above initial values, the recurrence relations (6.2.19) can be used to find all values e_j and $f_k, k = 1, 2, \cdots, N$. Once the system is in an upper triangular form, we can solve for $V_N, V_{N-1}, \cdots V_1$, successively by backward substitution, starting from $V_{N+1} = 0$.

The Thomas algorithm is a very efficient algorithm where the tridiagonal system (6.2.13) can be solved with 4 (add/subtract) and 6 (multiply/divide) operations per node point. Compared to the explicit schemes, it takes about twice the number of operations per time step. The solution of a tridiagonal system required by an implicit scheme does not add much computational complexity.

On the control of growth of roundoff errors, we observe that the calculations would be numerically stable provided that $|e_j| < 1$ so that error in V_{j+1} will not be magnified and propagated to V_j [see Eq. (6.2.17a)]. A set of sufficient conditions to guarantee $|e_j| < 1$ is given by

$$a_j > 0, b_j > 0, c_j > 0 \quad \text{and} \quad b_j > a_j + c_j. \tag{6.2.21}$$

Fortunately, the above conditions can be satisfied easily by the tridiagonal system (6.2.13) by the appropriate choices of $\Delta\tau$ and Δx in the Crank-Nicolson scheme.

6.2.3 Front fixing method and point relaxation method

In this subsection, we consider several numerical approaches for solving American option models using finite difference methods. The difficulties in the construction of numerical algorithms for solving American style option models arise from the unknown optimal exercise prices. First, we discuss the front fixing method, where a transformation of the independent variable is applied so that the free boundary associated with the optimal exercise prices is converted into a fixed boundary. The extension of the front fixing method to pricing of convertible bonds is reported by Zhu and Sun (1999). Recall that in the binomial/trinomial algorithm for pricing an American option, a dynamic programming procedure is applied at each node to determine whether the continuation value is less than the intrinsic value. If this is so, the intrinsic value is taken as the option value. We have difficulty in implementing the above dynamic programming procedure when an implicit scheme is employed since option values are obtained implicitly. We examine how the difficulty can be resolved by a point relaxation scheme. The third approach is called the penalty method, where we append an extra penalty term into the governing equation. In the limit, the resulting solution is guaranteed to satisfy the constraint that its value cannot be below the exercise payoff (see Problem 6.28).

Front fixing method

We consider the construction of the front fixing algorithm for finding the option value and the associated optimal exercise boundary $S^*(\tau)$ of an American put. For simplicity, we take the strike price to be unity. This is equivalent to normalize the underlying asset price and option value by the strike price. In the continuation region, the put value $P(S,\tau)$ satisfies the Black-Scholes equation

$$\frac{\partial P}{\partial \tau} - \frac{\sigma^2}{2}S^2\frac{\partial^2 P}{\partial S^2} - rS\frac{\partial P}{\partial S} + rP = 0, \quad \tau > 0, S^*(\tau) < S < \infty, \tag{6.2.22}$$

subject to the boundary conditions

$$P(S^*(\tau),\tau) = 1 - S^*(\tau), \frac{\partial P}{\partial S}(S^*(\tau),\tau) = -1, \lim_{S\to\infty} P(S,\tau) = 0, \tag{6.2.23a}$$

and initial condition

$$P(S,0) = 0 \quad \text{for} \quad S^*(0) < S < \infty, \tag{6.2.23b}$$

with $S^*(0) = 1$. We apply the transformation of the state variable: $y = \ln \frac{S}{S^*(\tau)}$ so that $y = 0$ at $S = S^*(\tau)$. Now, the free boundary $S = S^*(\tau)$ becomes the fixed boundary $y = 0$, hence the name of this method. In terms of the new independent variable y, the above governing equation becomes

$$\frac{\partial P}{\partial \tau} - \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial y^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\partial P}{\partial y} + rP = \frac{S^{*'}(\tau)}{S^*(\tau)}\frac{\partial P}{\partial y}, \tag{6.2.24}$$

subject to the new set of auxiliary conditions

$$P(0,\tau) = 1 - S^*(\tau), \frac{\partial P}{\partial y}(0,\tau) = -S^*(\tau), P(\infty,\tau) = 0, \tag{6.2.25a}$$

$$P(y,0) = 0 \quad \text{for } 0 < y < \infty. \tag{6.2.25b}$$

The non-linearity in the American put model is revealed by the non-linear term $\frac{S^{*'}(\tau)}{S^*(\tau)}\frac{\partial P}{\partial y}$. Along the boundary $y = 0$, we have the continuity of P, $\frac{\partial P}{\partial y}$ and $\frac{\partial P}{\partial \tau}$ so that $\frac{\partial^2 P}{\partial y^2}(0^+,\tau)$ observes the relation

$$\begin{aligned}\frac{\sigma^2}{2}\frac{\partial^2 P}{\partial y^2}(0^+,\tau) &= \frac{\partial}{\partial \tau}[1 - S^*(\tau)] - \left(r - \frac{\sigma^2}{2}\right)[-S^*(\tau)] \\ &\quad + r[1 - S^*(\tau)] - \frac{S^{*'}(\tau)}{S^*(\tau)}[-S^*(\tau)] \\ &= r - \frac{\sigma^2}{2}S^*(\tau). \end{aligned} \tag{6.2.26}$$

This derived relation is used to determine $S^*(\tau)$ once we have obtained $\frac{\partial^2 P}{\partial y^2}(0^+, \tau)$.

The direct Crank-Nicolson discretization of Eq. (6.2.24) would result in a non-linear algebraic system of equations for the determination of V_j^{n+1} due to the presence of the non-linear term $\frac{S^{*'}(\tau)}{S^*(\tau)} \frac{\partial P}{\partial y}$. To circumvent the difficulties while maintaining the same order of accuracy as that of the Crank-Nicholson scheme, we adopt a three-level scheme of the form

$$\frac{P_j^{n+1} - P_j^{n-1}}{2\Delta\tau} - \left[\frac{\sigma^2}{2} D_+ D_- + \left(r - \frac{\sigma^2}{2}\right) D_0 - r\right] \frac{P_j^{n+1} + P_j^{n-1}}{2}$$
$$= \frac{S_{n+1}^* - S_{n-1}^*}{2\Delta\tau S_n^*} D_0 P_j^n, \tag{6.2.27}$$

where S_n^* denotes the numerical approximation to $S^*(n\Delta\tau)$, while D_+, D_- and D_0 are discrete difference operators defined by

$$D_+ = \frac{1}{\Delta y}(E^1 - I), \; D_- = \frac{1}{\Delta y}(I - E^{-1}),$$
$$D_0 = \frac{1}{2\Delta y}(E^1 - E^{-1}). \tag{6.2.28}$$

Here, I denotes the identity operator and $E^i, i = -1, 1$, denotes the spatial shifting operator on a discrete function P_j, where $E^i P_j = P_{j+i}$.

The discretization of the value matching condition, smooth pasting condition and the boundary equation (6.2.26) lead to the following system of equations that relate P_{-1}^n, P_0^n, P_1^n and S_n^*:

$$P_0^n = 1 - S_n^* \tag{6.2.29a}$$
$$\frac{P_1^n - P_{-1}^n}{2\Delta y} = -S_n^* \tag{6.2.29b}$$
$$\frac{\sigma^2}{2}\left[\frac{P_1^n - 2P_0^n + P_{-1}^n}{\Delta y^2}\right] + \frac{\sigma^2}{2} S_n^* - r = 0. \tag{6.2.29c}$$

Here, P_{-1}^n is a fictitious value outside the computational domain. By eliminating P_{-1}^n, we obtain

$$P_1^n = \alpha - \beta S_n^*, \quad n \geq 1, \tag{6.2.30a}$$

where

$$\alpha = 1 + \frac{\Delta y^2}{\sigma^2 r} \quad \text{and} \quad \beta = \frac{1 + (1 + \Delta y)^2}{2}. \tag{6.2.30b}$$

Once P_1^n is known, we can find S_n^* using (6.2.30a) and P_0^n using (6.2.29a). For the boundary condition at the right end of the computational domain, we observe that the American put value tends to zero when S is sufficiently high. Therefore, we choose M to be sufficiently large such that we set $P_{M+1}^n = 0$ with sufficient accuracy.

Let $\mathbf{P}^n = (P_1^n \quad P_2^n \quad \cdots \quad P_M^n)^T$ and $\mathbf{e}_1 = (1 \quad 0 \quad \cdots \quad 0)^T$. By putting all the auxiliary conditions into the finite difference scheme (6.2.27), we would like to show how to calculate $\mathbf{P}^{n+1}$ from known values of $\mathbf{P}^n$ and $\mathbf{P}^{n-1}$. First, we define the following parameters

$$\begin{aligned} a &= \mu\sigma^2 + r\Delta\tau, \quad b = \frac{\mu}{2}\left[\sigma^2 - \Delta y\left(r - \frac{\sigma^2}{2}\right)\right], \\ c &= \frac{\mu}{2}\left[\sigma^2 + \Delta y\left(r - \frac{\sigma^2}{2}\right)\right], \end{aligned} \tag{6.2.31}$$

where $\mu = \dfrac{\Delta\tau}{\Delta y^2}$. Also, we define the tridiagonal matrix

$$A = \begin{pmatrix} a & -c & 0 & \cdots & \cdots & 0 \\ -b & a & -c & 0 & \cdots & 0 \\ 0 & -b & a & -c & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -b & a & -c \\ 0 & 0 & \cdots & 0 & -b & a \end{pmatrix}. \tag{6.2.32}$$

In terms of A, the finite difference scheme (6.2.27) can be expressed as

$$\begin{aligned} (I + A)\mathbf{P}^{n+1} = &(I - A)\mathbf{P}^{n-1} + bP_0^{n-1}\mathbf{e}_1 \\ &+ bP_0^{n+1}\mathbf{e}_1 + g^n D_0 \mathbf{P}^n, \quad n > 1, \end{aligned} \tag{6.2.33}$$

where $g^n = \dfrac{S_{n+1}^* - S_{n-1}^*}{S_n^*}$. By inverting the matrix $(I + A)$, Eq. (6.2.33) can be expressed as

$$\mathbf{P}^{n+1} = \mathbf{f}_1 + bP_0^{n+1}\mathbf{f}_2 + g^n\mathbf{f}_3 \tag{6.2.34}$$

where

$$\begin{aligned} \mathbf{f}_1 &= (I + A)^{-1}[(I - A)\mathbf{P}^{n-1} + bP_0^{n-1}\mathbf{e}_1], \\ \mathbf{f}_2 &= (I + A)^{-1}\mathbf{e}_1, \\ \mathbf{f}_3 &= (I + A)^{-1}D_0\mathbf{P}^n. \end{aligned} \tag{6.2.35}$$

Note that P_0^{n+1} and S_{n+1}^* can be expressed in terms of P_1^{n+1} using Eqs. (6.2.29a) and (6.2.30a).

Since Eq. (6.2.34) is a three-level scheme, we need $\mathbf{P}^1$ in addition to $\mathbf{P}^0$ to initialize the computation. To maintain overall second order accuracy, we employ the following two-step predictor-corrector technique to obtain $\mathbf{P}^1$:

$$\left(I + \frac{A}{2}\right)\widetilde{\mathbf{P}} = \left(I - \frac{A}{2}\right)\mathbf{P}^0 + \frac{b}{2}\widetilde{P}_0\mathbf{e}_1 + \widetilde{g}D_0\mathbf{P}^0,$$

$$\left(I + \frac{A}{2}\right)\mathbf{P}^1 = \left(I - \frac{A}{2}\right)\mathbf{P}^0 + \frac{b}{2}P_0^1\mathbf{e}_1 + g^1 D_0\left(\frac{\widetilde{\mathbf{P}} + \mathbf{P}^0}{2}\right), \quad (6.2.36)$$

where the provisional values and g^1 are given by

$$\widetilde{P}_0 = 1 - \widetilde{S}_0^*, \quad \widetilde{S}_0^* = \frac{\alpha - \widetilde{P}_1}{\beta},$$

$$\widetilde{g} = \frac{\widetilde{S}_0^* - S_0^*}{S_0^*} \quad \text{and} \quad g^1 = \frac{S_1^* - S_0^*}{\frac{\widetilde{S}_0^* + S_0^*}{2}}. \quad (6.2.37)$$

Projected successive-over-relaxation method
Consider an implicit finite difference scheme in the form

$$a_{-1}V_{j-1} + a_0V_j + a_1V_{j+1} = d_j, \quad j = 1, 2, \cdots, N, \quad (6.2.38)$$

where the superscript "$n+1$" is omitted for brevity, and d_j represents the known quantities. Instead of solving the tridiagonal system by direct elimination (Thomas algorithm), one may choose to use the iterative method. The Gauss-Seidel iterative procedure produces the k^{th} iterate of V_j by

$$\begin{aligned} V_j^{(k)} &= \frac{1}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_1V_{j+1}^{(k-1)}\right) \\ &= V_j^{(k-1)} + \frac{1}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_0V_j^{(k-1)} - a_1V_{j+1}^{(k-1)}\right), \end{aligned} \quad (6.2.39)$$

where the last term in the above equation represents the correction made on the $(k-1)^{\text{th}}$ iterate of V_j. To accelerate the rate of convergence of the iteration, we multiply the correction term by a relaxation parameter ω. The corresponding iterative procedure becomes

$$V_j^{(k)} = V_j^{(k-1)} + \frac{\omega}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_0V_j^{(k-1)} - a_1V_{j+1}^{(k-1)}\right),$$
$$0 < \omega < 2. \quad (6.2.40)$$

This procedure is called the *successive-over-relaxation*. As a necessary condition for convergence, the relaxation parameter ω must be chosen between 0 and 2.

Let h_j denote the intrinsic value of the American option at the j^{th} node. To incorporate the constraint that the option value must be above the intrinsic value, the dynamic programming procedure in combination with the above relaxation procedure is then designed to be

$$V_j^{(k)} = \max\left(V_j^{(k-1)} + \frac{\omega}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_0 V_j^{(k-1)} - a_1 V_{j+1}^{(k-1)}\right), h_j\right) \tag{6.2.41}$$

We perform a sufficient number of iterations until the following termination criterion is met:

$$\sqrt{\sum_{j=1}^{N}\left(V_j^{(k)} - V_j^{(k-1)}\right)^2} < \epsilon, \quad \epsilon \text{ is some small tolerance value.}$$

The convergent value $V_j^{(k)}$ is then taken to be the numerical solution for V_j. The above iterative scheme is called the *projected successive-over-relaxation method*.

6.2.4 Truncation errors and order of convergence

The local truncation error of a given numerical scheme is obtained by substituting the exact solution of the continuous problem into the numerical scheme. Let $V(j\Delta x, n\Delta\tau)$ denote the exact solution of the continuous Black-Scholes equation. We illustrate the procedure of finding the local truncation error of the explicit FTCS scheme by substituting the exact solution into the explicit scheme. The local truncation error at the node point $(j\Delta x, n\Delta\tau)$ is given by

$$\begin{aligned} &T(j\Delta x, n\Delta\tau) \\ &= \frac{V(j\Delta x,(n+1)\Delta\tau) - V(j\Delta x, n\Delta\tau)}{\Delta\tau} \\ &\quad - \frac{\sigma^2}{2}\frac{V((j+1)\Delta x, n\Delta\tau) - 2V(j\Delta x, n\Delta\tau) + V((j-1)\Delta x, n\Delta\tau)}{\Delta x^2} \\ &\quad - \left(r - \frac{\sigma^2}{2}\right)\frac{V((j+1)\Delta x, n\Delta\tau) - V((j-1)\Delta x, n\Delta\tau)}{2\Delta x} \\ &\quad + rV(j\Delta x, n\Delta\tau). \end{aligned} \tag{6.2.42}$$

We then expand each term by peforming the Taylor expansion at the node point $(j\Delta x, n\Delta\tau)$. After some cancellation of terms, we obtain

$$\begin{aligned} &T(j\Delta x, n\Delta\tau) \\ &= \frac{\partial V}{\partial\tau}(j\Delta x, n\Delta\tau) + \frac{\Delta\tau}{2}\frac{\partial^2 V}{\partial\tau^2}(j\Delta x, n\Delta\tau) + O\left(\Delta\tau^2\right) \\ &\quad - \frac{\sigma^2}{2}\left[\frac{\partial^2 V}{\partial x^2}(j\Delta x, n\Delta\tau) + \frac{\Delta x^2}{12}\frac{\partial^4 V}{\partial x^4}(j\Delta x, n\Delta\tau) + O(\Delta x^4)\right] \\ &\quad - \left(r - \frac{\sigma^2}{2}\right)\left[\frac{\partial V}{\partial x}(j\Delta x, n\Delta\tau) + \frac{\Delta x^2}{3}\frac{\partial^3 V}{\partial x^3}(j\Delta x, n\Delta\tau) + O(\Delta x^4)\right] \\ &\quad + rV(j\Delta x, n\Delta\tau). \end{aligned} \tag{6.2.43}$$

Since $V(j\Delta x, n\Delta\tau)$ satisfies the Black-Scholes equation, this leads to

$$\begin{aligned}T(j\Delta x, n\Delta\tau) = &\frac{\Delta\tau}{2}\frac{\partial^2 V}{\partial\tau^2}(j\Delta x, n\Delta\tau) - \frac{\sigma^2}{24}\Delta x^2\frac{\partial^4 V}{\partial x^4}(j\Delta x, n\Delta\tau)\\ &-\left(r-\frac{\sigma^2}{2}\right)\frac{\Delta x^2}{3}\frac{\partial^3 V}{\partial x^3}(j\Delta x, n\Delta\tau) + O(\Delta\tau^2)\\ &+ O(\Delta x^4).\end{aligned} \tag{6.2.44}$$

The local truncation error measures the discrepancy that the continuous solution does not satisfy the numerical scheme at the node point.

A necessary condition for the convergence of the numerical solution to the continuous solution is that the local truncation error of the numerical scheme must tend to zero for vanishing stepwidth and time step. In this case, the numerical scheme is said to be *consistent*. The *order of accuracy* of a scheme is defined to be the order in powers of Δx and $\Delta\tau$ in the leading truncation error terms. Suppose the leading truncation terms are $O(\Delta\tau^k, \Delta x^m)$, then the numerical scheme is said to be k^{th} order time accurate and m^{th} order space accurate. From Eq. (6.2.44), we observe that the explicit FTCS scheme is first order time accurate and second order space accurate. Suppose we choose $\Delta\tau$ to be the same order as Δx^2, that is, $\Delta\tau = \lambda\Delta x^2$ for some finite constant λ (recall that the same relation between $\Delta\tau$ and Δx has been adopted by the trinomial scheme), then the leading truncation error terms in Eq. (6.2.44) become $O(\Delta\tau)$.

Using a similar technique of performing Taylor expansion, one can show that all versions of the binomial scheme are first order time accurate (recall that $\Delta\tau$ and Δx are dependent in binomial schemes). This is not surprising since we have done similar error analysis in Sec. 6.1, though the converse argument has been used. In the earlier analysis, we find to what extent the numerical solution from the binomial scheme satisfies the continuous Black-Scholes equation. Either approach gives the same conclusion on the order of accuracy.

For the implicit Crank-Nicolson scheme, it can be shown that it is second order time accurate and second order space accurate (see Problem 6.21). The highest order of accuracy that can be achieved for a two-level six-point scheme is known to be $O(\Delta\tau^2, \Delta x^4)$ (see the compact scheme given in Problem 6.22). With regard to accuracy consideration, higher order schemes should be preferred over lower order schemes.

Suppose the leading truncation error terms of a numerical scheme are $O(\Delta\tau^m)$, m is some positive integer, one can show from more advanced theoretical analysis that the numerical solution $V_j^n(\Delta\tau)$ using time step $\Delta\tau$ has the asymptotic expansion of the form

$$V_j^n(\Delta\tau) = V_j^n(0) + K\Delta\tau^m + O(\Delta\tau^{m+1}), \tag{6.2.45a}$$

where $V_j^n(0)$ is visualized as the exact continuous solution obtained in the limit $\Delta\tau \to 0$, and K is some constant independent of $\Delta\tau$. Suppose we perform two numerical calculations using time step $\Delta\tau$ and $\frac{\Delta\tau}{2}$ successively, it is easily seen that

$$V_j^n(0) - V_j^n(\Delta\tau) \approx 2^m \left[V_j^n(0) - V_j^n\left(\frac{\Delta\tau}{2}\right)\right]. \qquad (6.2.45b)$$

That is, the error in the numerical solution of a m^{th}-order time accurate scheme is reduced by a factor of $\frac{1}{2^m}$ when we reduce the time step by a factor of $\frac{1}{2}$.

6.2.5 Numerical stability and oscillation phenomena

A numerical scheme must be consistent in order that the numerical solution converges to the exact solution of the underlying differential equation. However, consistency is only a necessary but not sufficient condition for convergence. The roundoff errors incurred during numerical calculations may lead to the blow up of the solution and erode the whole computation. Besides the analysis of the truncation error, it is necessary to analyze the stability properties of a numerical scheme. Loosely speaking, a scheme is said to be stable if roundoff errors are not amplified in numerical computation. For a linear evolutionary differential equation, like the Black-Scholes equation, the *Lax Equivalence Theorem* states that stability is the necessary and sufficient condition for the convergence of a consistent difference scheme.

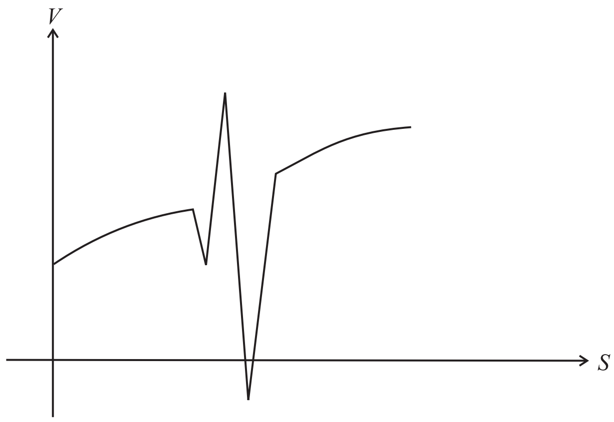

Fig. 6.6 Spurious oscillations in numerical solution of option price.

Another undersirable feature in the behaviors of the finite difference solution is the occurrence of spurious oscillations. It is possible to generate negative option values even if the scheme is stable (see Fig. 6.6). The oscillation phenomena in the numerical calculations of barrier and Asian option models are discussed in the papers by Zvan *et al.* (1998, 2000).

Fourier method of stability analysis

There exists a huge body of literature on stability analysis of numerical schemes, and different notions of stability have also been defined. Here, we only discuss the *Fourier method of stability analysis*. The Fourier method is based on the assumption that the numerical scheme admits a solution of the form

$$V_j^n = A^n(k)e^{ikj\Delta x}, \tag{6.2.46}$$

where k is the wavenumber and $i = \sqrt{-1}$. The von Neumann stability criterion examines the growth of the above Fourier component. Substituting Eq. (6.2.46) into the two-level six-point scheme (6.2.11), we obtain

$$G(k) = \frac{A^{n+1}(k)}{A^n(k)} = \frac{b_1 e^{ik\Delta x} + b_0 + b_1 e^{-ik\Delta x}}{a_1 e^{ik\Delta x} + a_0 + a_{-1}e^{-ik\Delta x}}, \tag{6.2.47}$$

where $G(k)$ is the amplification factor which governs the growth of the Fourier component over one time period. The strict von Neumann stability condition is given by

$$|G(k)| \le 1, \tag{6.2.48}$$

for $0 \le k\Delta x \le \pi$. Henceforth, we write $\beta = k\Delta x$.

We now apply the Fourier stability analysis to study the stability properties of the Cox-Ross-Rubinstein binomial scheme and the implicit Crank-Nicolson scheme.

Cox-Ross-Rubinstein binomial scheme

The corresponding amplification factor of the Cox-Ross-Rubinstein binomial scheme is

$$G(\beta) = (\cos\beta + iq\sin\beta)e^{-r\Delta\tau}, \quad q = 2p - 1. \tag{6.2.49}$$

The von Neumann stability condition requires

$$|G(\beta)|^2 = \left[1 + (q^2-1)\sin^2\beta\right]e^{-2r\Delta\tau} \le 1, \quad 0 \le \beta \le \pi. \tag{6.2.50}$$

When $0 \le p \le 1$, we have $|q| \le 1$ so that $|G(\beta)| \le 1$ for all β. Under this condition, the scheme is guaranteed to be stable in the von Neumann sense. However, suppose we choose $p > 1$, the binomial scheme becomes unstable when

$$p = \frac{e^{r\Delta\tau} - e^{-\sigma\sqrt{\Delta\tau}}}{e^{\sigma\sqrt{\Delta\tau}} - e^{-\sigma\sqrt{\Delta\tau}}} > \frac{e^{r\Delta\tau}+1}{2}. \tag{6.2.51}$$

It seems intractable to solve for $\Delta\tau$ explicitly in terms of r and σ for deducing a constraint on $\Delta\tau$ as a stability condition. However, at least we have deduced a sufficient condition for von Neumann stability: non-occurrence of negative probability values in the binomial scheme.

Crank-Nicolson scheme
The corresponding amplification factor of the Crank-Nicolson scheme is found to be

$$G(\beta) = \frac{1 - \sigma^2 \frac{\Delta\tau}{\Delta x^2} \sin^2 \frac{\beta}{2} + \left(r - \frac{\sigma^2}{2}\right) \frac{\Delta\tau}{2\Delta x} i \sin\beta - \frac{r}{2}\Delta\tau}{1 + \sigma^2 \frac{\Delta\tau}{\Delta x^2} \sin^2 \frac{\beta}{2} - \left(r - \frac{\sigma^2}{2}\right) \frac{\Delta\tau}{2\Delta x} i \sin\beta + \frac{r}{2}\Delta\tau}. \tag{6.2.52}$$

The von Neumann stability condition requires

$$|G(\beta)|^2 = \frac{\left(1 - \sigma^2 \frac{\Delta\tau}{\Delta x^2} \sin^2 \frac{\beta}{2} - \frac{r}{2}\Delta\tau\right)^2 + \left(r - \frac{\sigma^2}{2}\right)^2 \frac{\Delta\tau^2}{4\Delta x^2} \sin^2 \beta}{\left(1 + \sigma^2 \frac{\Delta\tau}{\Delta x^2} \sin^2 \frac{\beta}{2} + \frac{r}{2}\Delta\tau\right)^2 + \left(r - \frac{\sigma^2}{2}\right)^2 \frac{\Delta\tau^2}{4\Delta x^2} \sin^2 \beta} \leq 1,$$
$$0 \leq \beta \leq \pi. \tag{6.2.53}$$

It is easily seen that the above condition is satisfied for any choices of $\Delta\tau$ and Δx. Hence, the Crank-Nicolson scheme is unconditionally stable. In other words, numerical stability (in von Neumann sense) is ensured without any constraint on the choice of $\Delta\tau$.

The implicit Crank-Nicolson scheme is observed to have second order temporal accuracy and unconditional stability. Also, the implementation of the numerical computation can be quite efficient with the use of the Thomas algorithm. Apparently, practitioners should favor the Crank-Nicolson scheme over other conditionally stable and first order time accurate explicit schemes. Unfortunately, the numerical accuracy of the finite difference solution can be much deteriorated due to non-smooth property of the terminal payoff function in most option models. The issues of implementation of the auxiliary conditions in option pricing using finite difference schemes are discussed in Sec. 6.2.6.

Spurious oscillations of numerical solution
It is relatively easy to find the sufficient conditions for non-appearance of spurious oscillations in the numerical solution of a two-level explicit scheme. The following theorem reveals the relation between the signs of the coefficients in the explicit scheme and spurious oscillations of the computed solution (Kwok and Lau, 2001b).

Theorem
Suppose the coefficients in the two-level explicit scheme (6.2.4) are all non-negative, and the initial values are bounded, that is, $\max_j |V_j^0| \leq M$ for some constant M; then

$$\max_j |V_j^n| \le M \quad \text{for all} \quad n \ge 1. \tag{6.2.54}$$

The proof of the above theorem is quite straightforward. From the explicit scheme, we deduce that

$$|V_j^{n+1}| \le |b_{-1}|\,|V_{j-1}^n| + |b_0|\,|V_j^n| + |b_1|\,|V_{j+1}^n|, \tag{6.2.55a}$$

and so

$$\max_j |V_j^{n+1}| \le b_{-1} \max_j |V_{j-1}^n| + b_0 \max_j |V_j^n| + b_1 \max_j |V_{j+1}^n| \tag{6.2.55b}$$

since b_{-1}, b_0 and b_1 are non-negative. Let E^n denote $\max_j |V_j^n|$, Ineq. (6.2.55b) can be expressed as

$$E^{n+1} \le b_{-1}E^n + b_0 E^n + b_1 E^n = E^n \tag{6.2.56}$$

since $b_{-1} + b_0 + b_1 = 1$. Deductively, we obtain

$$E^n \le E^{n-1} \le \cdots \le E^0 = \max_j |V_j^0| = M. \tag{6.2.57}$$

What happens when one or more of the coefficients of the explicit scheme become negative? For example, we take $b_0 < 0, b_{-1} > 0$ and $b_1 > 0$, and let $V_0^0 = \varepsilon > 0$ and $V_j^0 = 0, j \ne 0$. At the next time level, $V_{-1}^1 = b_1\varepsilon, V_0^1 = b_0\varepsilon$ and $V_1^1 = b_{-1}\varepsilon$, where the sign of V_j^1 alternates with j. This alternating sign property can be shown to persist at all later time levels. In this way, we deduce that

$$|V_j^{n+1}| = b_{-1}|V_{j-1}^n| - b_0|V_j^n| + b_1|V_{j+1}^n|. \tag{6.2.58}$$

We sum over all values of j of the above equation and let $S^n = \sum_j |V_j^n|$. As a result, we obtain

$$S^{n+1} = (b_{-1} - b_0 + b_1)S^n = (1 - 2b_0)S^n. \tag{6.2.59}$$

Note that $1 - 2b_0 > 1$ since $b_0 < 0$. Deductively, we obtain

$$S^n = (1 - 2b_0)^n S_0 = (1 - 2b_0)^n \varepsilon, \tag{6.2.60}$$

and as $n \to \infty, S^n \to \infty$. The solution values oscillate in signs at neighboring nodes, and the oscillation amplitudes grow with increasing number of time steps.

6.2.6 Numerical approximations of auxiliary conditions

The errors observed in the finite difference solution may arise from various sources. The major source is the truncation error, which stems from the difference approximation of the differential operators. Another source comes from the numerical approximation of the auxiliary conditions, referring to the terminal payoff in all types of options and boundary conditions in path dependent options. It is commonly observed that numerical option values obtained from trinomial or finite difference calculations exhibit wavy or erratic pattern of convergence to the continuous solutions. Heston and Zhou (2000) illustrate from their numerical experiments that the rate of convergence of binomial calculations fluctuate between $O(\sqrt{\Delta t})$ and $O(\Delta t)$. Due to lack of smooth convergence, extrapolation technique for the enhancement of the rate of convergence cannot be routinely applied to numerical option values. In this subsection, we present several smoothness-enhancement techniques for dealing with discontinuity and non-differentiability of the terminal payoff function and proper treatment of numerical boundary conditions which are associated with barrier and lookback features (the path dependent features can be continuously or discretely monitored).

Smoothing of discontinuities in terminal payoff functions

Most terminal payoff function of options have some form of discontinuity (like binary payoff) or non-differentiability (like call or put payoff). Quantization error arises since the payoff function is sampled at discrete node points. Several smoothing techniques have been proposed in the literature. Heston and Zhou (2000) propose to set the payoff value at node in the computational mesh by the average of the payoff function over the surrounding node cells rather than sampled at the node point. Let $V_T(S)$ denote the terminal payoff function. The payoff value at node S_j is given by

$$V_j^0 = \frac{1}{\Delta S} \int_{S_j - \frac{\Delta S}{2}}^{S_j + \frac{\Delta S}{2}} V_T(S)\, dS \tag{6.2.61}$$

instead of $V_T(S_j)$. Take the call payoff $\max(S - X, 0)$ as an example. If the strike price X falls exactly on a node point, then $V_T(S_j) = 0$ while the cell-averaged value is $\Delta S/8$. In their binomial calculations, Heston and Zhou (2000) find that averaging the payoff for vanilla European and American calls provide a more smooth convergence that subsequently allows for the application of extrapolation for convergence enhancement. Another simple technique is the method of node positioning. Tavella and Randall (2002) propose to place the strike price halfway between node points. The third technique is called Black-Scholes approximation, which is useful for pricing American options and exotic options for which the Black-Scholes solution is a good approximation at time close to expiry. The trick is to use the

Black-Scholes values along the first time level and proceed with usual finite difference calculations for subsequent time levels.

More advanced methods for minimizing the quantization errors in higher order schemes have also been studied. Pooley *et al.* (2003) show that if discontinuous terminal payoff is present, the expected second order convergence of the Crank-Nicolson scheme cannot be realized. They manage to develop elaborate techniques that can be used to recover the quadratic rate of convergence. Raahauge (2005) proposes some transformation technique to transform the original ill-conditional pricing problem into a well behaved numerical problem so that high order numerical methods can be implemented effectively.

Barrier options

The two major factors that lead to deterioration of numerical accuracy in barrier option calculations are (i) positioning of the nodes relative to the barrier, (ii) proximity of the initial asset price to the barrier.

Several papers have reported that better numerical accuracy can be achieved if the barrier is placed to pass through a layer of nodes for the continuously monitored barrier, and located halfway between two layers of nodes for the discretely monitored barrier. Heuristic arguments that explain why these choices of positioning achieve better numerical accuracy can be found in Kwok and Lau's paper (2001b). To remedy the proximity problem, Figlewski and Gao (1999) suggest to construct fine meshes near the barrier to improve the level of accuracy. However, Boyle and Tian (1998) show that the application of spline interpolation of option values at three adjacent nodes is a simple method to resolve the problem of dealing with the proximity issue. For implicit schemes, "initial asset price close to the barrier" is not an issue since the response to boundary conditions are felt almost instantaneously across the entire solution in implicit scheme calculations (Zvan *et al.*, 2000).

Lookback options

It is relatively straightforward to price lookback options using forward shooting grid approach (see Problem 6.17). For floating strike lookback options, by applying appropriate choices of similarity variables, the pricing formulation reduces to the form similar to that of usual one-asset option models, except that Neumann boundary condition appears at one end of the domain of the lookback option model. Let $c(S, m, t)$ denote the price of a continuously monitored European floating strike lookback call option, where m is the realized minimum asset price from T_0 to t. The terminal payoff at time T of the lookback call is given by

$$c(S, m, T) = S - m. \tag{6.2.62}$$

Recall that $S \geq m$ and the boundary condition at $S = m$ is given by

$$\frac{\partial c}{\partial m} = 0 \quad \text{at} \quad S = m. \tag{6.2.63}$$

We choose the following set of similarity variables:

$$x = \ln \frac{S}{m} \quad \text{and} \quad V(x, \tau) = \frac{c(S, m, t)}{S} e^{-q\tau}, \tag{6.2.63}$$

where $\tau = T - t$, then the Black-Scholes equation for c is tranformed into the following equation for V.

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2}{2} \frac{\partial^2 V}{\partial x^2} + \left(r - q + \frac{\sigma^2}{2} \right) \frac{\partial V}{\partial x}, \quad x > 0, \tau > 0. \tag{6.2.64}$$

Note that $S > m$ corresponds to $x > 0$. The terminal payoff condition becomes the following initial condition

$$V(x, 0) = 1 - e^{-x}, \quad x > 0. \tag{6.2.65}$$

The boundary condition at $S = m$ becomes the Neumann condition

$$\frac{\partial V}{\partial x}(0, \tau) = 0. \tag{6.2.66}$$

Suppose we discretize the governing equation using the FTCS scheme, we obtain

$$V_j^{n+1} = \left[\frac{\alpha + \mu}{2} V_{j+1}^n + (1 - \alpha) V_j^n + \frac{\alpha - \mu}{2} V_{j-1}^n \right], j = 1, 2, \cdots \tag{6.2.67}$$

where $\alpha = \left(r - q + \frac{\sigma^2}{2} \right) \frac{\Delta \tau}{\Delta x}$ and $\mu = \sigma^2 \frac{\Delta \tau}{\Delta x^2}$. For continuously monitored lookback option mdoel, we place the reflecting boundary $x = 0$ (corresponding to the Neumann boundary condition) along a layer of nodes, where the node $j = 0$ corresponds to $x = 0$. To approximate the Neumann boundary condition at $x = 0$, we use the centered difference

$$\left. \frac{\partial V}{\partial x} \right|_{x=0} \approx \frac{V_1^n - V_{-1}^n}{2 \Delta x}, \tag{6.2.68}$$

where V_{-1}^n is the option value at a fictitious node one cell to the left of node $j = 0$. By setting $j = 0$ in Eq. (6.2.27) and applying the approximation of the Neumann condition: $V_1^n = V_{-1}^n$, we obtain

$$V_0^{n+1} = \alpha V_1^n + (1 - \alpha) V_0^n. \tag{6.2.69}$$

Numerical results obtained from the above scheme demonstrate $O(\Delta t)$ rate of convergence (Kwok and Lau, 2001b). However, suppose forward difference is used to approximate $\left. \frac{\partial V}{\partial x} \right|_{x=0}$ so that the Neumann boundary condition is approximated by $V_0^n = V_{-1}^n$ (Cheuk and Vorst, 1997), then the order of convergence reduces to $O(\sqrt{\Delta t})$ only. Also, when the nodes are not chosen to align along the reflecting boundary, erratic convergence behaviors of the numerical results are observed. Problem 6.25 illustrates the failure of a naive treatment of the reflecting boundary condition of a lookback put option and Problem 6.26 demonstrates another approach of constructing the numerical boundary condition approximating the Neumann boundary condition.

It is quite tricky to price discretely sampled lookback options since the Neumann condition is applied only on those time steps that correspond to monitoring instants. Discussion of the construction of effective pricing algorithms can be found in the papers by Andreasen (1998) and Kwok and Lau (2001b),

6.3 Monte Carlo simulation

We have observed that a wide class of derivative pricing problems come down to the evaluation of the following expectation functional

$$Ef[Z(T;t,z)].$$

Here, Z denotes the stochastic process that describes the price evolution of one or more underlying financial variables such as asset prices and interest rates, under the respective risk neutral probability distributions. The process Z has the initial value z at time t, and the function f specifies the value of the derivative at the expiration time T.

As the third alternative other than the binomial and finite difference methods for the numerical valuation of derivative pricing problems, the Monte Carlo simulation has been proven to be a powerful and versatile technique. The Monte Carlo method is basically a numerical procedure for estimating the expected value of a random variable, and so it leads itself naturally to derivative pricing problems represented as expectations. The simulation procedure involves generating random variables with a given probability density and using the law of large numbers to take the average of these values as an estimate of the expected value of the random variable. In the context of derivative pricing, the Monte Carlo procedure involves the following steps.

(i) Simulate sample paths of the underlying state variables in the derivative model such as asset prices and interest rates over the life of the derivative, according to the risk neutral probability distributions.

(ii) For each simulated sample path, evaluate the discounted cash flows of the derivative.

(iii) Take the sample average of the discounted cash flows over all sample paths.

As an example, we consider the valuation of a European vanilla call option to illustrate the Monte Carlo procedure. The numerical procedure requires the computation of the expected payoff of the call option at expiry, $E_t[\max(S_T - X, 0)]$, and discounted to the present value at time t, namely, $e^{-r(T-t)}E_t[\max(S_T - X, 0)]$. Here, S_T is the asset price at expiration time

T and X is the strike price. Assuming lognormal distribution for the asset price movement, the price dynamics under the risk neutral measure is given by [see Eq. (2.4.5)]

$$\frac{S_{t+\triangle t}}{S_t} = e^{\left(r-\frac{\sigma^2}{2}\right)\triangle t+\sigma\epsilon\sqrt{\triangle t}}, \tag{6.3.1}$$

where $\triangle t$ is the time step, σ is the volatility and r is the riskless interest rate. Here, ϵ denotes a normally distributed random variable with zero mean and unit variance, and so $\sigma\epsilon\sqrt{\triangle t}$ represents a discrete approximation to an increment in the Wiener process of the asset price with volatility σ in time increment $\triangle t$. The random number ϵ can be generated in most computer programming languages, and because of its randomness, it assumes a different value in each generation run. Suppose these are N time steps between the current time t and expiration time T, where $\triangle t = (T-t)/N$. The numerical procedure given in Eq. (6.3.1) is repeated N times to simulate the price path from S_t to $S_T = S_{t+N\triangle t}$. The call price corresponding to this particular simulated asset price path is then computed using the discounted formula

$$c = e^{-r(T-t)} \max(S_T - X, 0). \tag{6.3.2}$$

This completes one sample iteration of the Monte Carlo simulation for this European call option model.

After repeating the above simulation for a sufficiently large number of runs, the expected call value is obtained by computing the average of the estimates of the call value found in the sample simulation. Also, the standard deviation of the estimate of the call value can be found. Let c_i denote the estimate of the call value obtained in the i^{th} simulation and M be the total number of simulation runs. The expected call value is given by

$$\hat{c} = \frac{1}{M}\sum_{i=1}^{M} c_i, \tag{6.3.3}$$

and the variance of the estimate is computed by

$$\hat{s}^2 = \frac{1}{M-1}\sum_{i=1}^{M}(c_i - \hat{c})^2. \tag{6.3.4}$$

For a sufficiently large value of M, the distribution

$$\frac{\hat{c}-c}{\sqrt{\frac{\hat{s}^2}{M}}}, \quad c \text{ is the true call value,}$$

tends to the standard normal distribution. Note that the standard deviation of $\hat{c}$ is equal to $\hat{s}/\sqrt{M}$ and so the confidence limits of estimation can be reduced by increasing the number of simulation runs M. The appearance of M as the factor $1/\sqrt{M}$ implies that the reduction of the standard deviation by a factor of 10 will require an increase of the number of simulation runs by 100 times.

One major advantage of the Monte Carolo method is that the error is independent of the dimension of the option problem. Another advantage is its ease to accommodate complicated payoff in an option model. For example, the terminal payoff of an Asian option depends on the average of the asset price over certain time interval while that of a lookback option depends on the extremum value of the asset price over some period of time. It is quite straightforward to obtain the average or extremum value in the simulated price path in each simulated path. The main drawback of the Monte Carlo simulation is the demand for a large number of simulation trials in order to achieve a high level of accuracy. This makes the simulation method less competitive compared to the binomial method and finite difference algorithms when analytic properties of the corresponding pricing model of an option are better known and formulated. However, viewing from another perspective, practitioners dealing with a newly invented option may obtain an estimate of its price using the Monte Carlo approach through routine simulation, rather than risking themselves in the construction of an analytic pricing model for the new option.

The efficiency of a Monte Carlo simulation can be greatly enhanced through the use of various variance reduction techniques (Boyle *et al.*, 1997), some of which are presented as follows.

6.3.1 Variance reduction techniques

It seems greatly desirable to reduce the variance $\hat{s}^2$ of the estimate so that a significant reduction in the number of simulation trials M may result. The two most common techniques of variance reduction are the *antithetic variates method* and the *control variate method*.

First, we would like to describe how to assess the effectiveness of a variance reduction technique from the perspective of computational efficiency. Suppose W_T is the total amount of computational work units available to generate an estimate of the value of an option V. Assume that there are two methods for generating the Monte Carlo estimates for the option value, requiring W_1 and W_2 units of computation work respectively for each simulation run. For simplicity, assume W_T is divisible by both W_1 and W_2. Let $V_i^{(1)}$ and $V_i^{(2)}$ denote the estimator of V in the i^{th} simulation using Methods 1 and 2, respectively, and their respective standard deviations are σ_1 and σ_2. The sample means for estimating V from the two methods using W_T amount of work are, respectively,

$$\frac{W_1}{W_T} \sum_{i=1}^{W_T/W_1} V_i^{(1)} \quad \text{and} \quad \frac{W_2}{W_T} \sum_{i=1}^{W_T/W_2} V_i^{(2)}.$$

By the law of large numbers, the above two estimators are approximately normally distributed with mean V and their respective standard deviations are

$$\sigma_1\sqrt{\frac{W_1}{W_T}} \quad \text{and} \quad \sigma_2\sqrt{\frac{W_2}{W_T}}.$$

Hence, the first method would be preferred over the second one provided that

$$\sigma_1^2 W_1 < \sigma_2^2 W_2. \tag{6.3.5}$$

Alternatively speaking, a lower variance estimator is preferred only if the variance ratio σ_1^2/σ_2^2 is less than the work ratio W_2/W_1, when the aspect of computational efficiency is taken into account.

Antithetic variates method

Suppose $\{\epsilon^{(i)}\}$ denotes the independent samples from the standard normal distribution for the i^{th} simulation run of the asset price path so that

$$S_T^{(i)} = S_t\, e^{\left(r-\frac{\sigma^2}{2}\right)(T-t)+\sigma\sqrt{\triangle t}\sum_{j=1}^{N}\epsilon_j^{(i)}}, \quad i = 1, 2, \cdots, M, \tag{6.3.6}$$

where $\triangle t = \dfrac{T-t}{N}$ and M is the total number of simulation runs. Note that $\epsilon_j^{(i)}$ is randomly sampled from the standard normal distribution. From Eqs. (6.3.2–3), an unbiased estimator of the price of a European call option with strike price X is given by

$$\hat{c} = \frac{1}{M}\sum_{i=1}^{M} c_i = \frac{1}{M}\sum_{i=1}^{M} e^{-r(T-t)}\max(S_T^{(i)} - X, 0). \tag{6.3.7a}$$

We observe that if $\{\epsilon^{(i)}\}$ has a standard normal distribution, so does $\{-\epsilon^{(i)}\}$, and the simulated price $\tilde{S}_T{}^{(i)}$ obtained from Eq. (6.3.6) using $\{-\epsilon^{(i)}\}$ is also a valid sample from the terminal asset price distribution. A new unbiased estimator of the call price can be obtained from

$$\tilde{c} = \frac{1}{M}\sum_{i=1}^{M} \tilde{c}_i = \frac{1}{M}\sum_{i=1}^{M} e^{-r(T-t)}\max(\tilde{S}_T^{(i)} - X, 0). \tag{6.3.7b}$$

Normally we would expect c_i and $\tilde{c}_i$ to be negatively correlated, that is, if one estimate overshoots the true value, the other estimate downshoots the true value. It seems sensible to take the average of these two estimates. Indeed, we take the antithetic variates estimate to be

$$\bar{c}_{AV} = \frac{\hat{c} + \tilde{c}}{2}. \tag{6.3.8}$$

Considering the aspect of computational efficiency as governed by inequality (6.3.5), it can be shown that the antithetic variates method improves efficiency provided that $\text{cov}(c_i, \tilde{c}_i) \leq 0$ (see Problem 6.29).

Control variate method

The control variate method is applicable when there are two similar options, A and B. Option A is the one whose price value is desired, while option B is similar to option A in nature but its analytic price formula is available. Let V_A and V_B denote the true value of option A and option B respectively, and let $\hat{V}_A$ and $\hat{V}_B$ denote the respective estimated value of option A and option B using the Monte Carlo simulation. How does the knowledge of V_B and $\hat{V}_B$ help improve the estimate of the value of option A beyond the available estimate $\hat{V}_A$?

The control variate method aims to provide a better estimate of the value of option A using the formula

$$\hat{V}_A^{cv} = \hat{V}_A + (V_B - \hat{V}_B), \tag{6.3.9}$$

where the error $V_B - \hat{V}_B$ is used as a control in the estimation of V_A. To justify the method, we consider the following relation between the variances of the above quantities

$$\text{var}\left(\hat{V}_A^{cv}\right) = \text{var}(\hat{V}_A) + \text{var}(\hat{V}_B) - 2\,\text{cov}(\hat{V}_A, \hat{V}_B), \tag{6.3.10}$$

so that

$$\text{var}(\hat{V}_A^{cv}) < \text{var}(\hat{V}_A) \text{ provided that } \text{var}(\hat{V}_B) < 2\,\text{cov}(\hat{V}_A, \hat{V}_B). \tag{6.3.11}$$

Hence, the control variate technique reduces the variance of the estimator of V_A when the covariance between $\hat{V}_A$ and $\hat{V}_B$ is large. This is true when the two options are strongly correlated. In terms of computational efforts, we need to compute two estimates $\hat{V}_A$ and $\hat{V}_B$. However, if the underlying asset price paths of the two options are identical, then there is only slight additional work to evaluate $\hat{V}_B$ along with $\hat{V}_A$ on the same set of simulated price paths.

To facilitate the more optimal use of the control $V_B - \hat{V}_B$, we define the control variate estimate to be

$$\hat{V}_A^{\beta} = \hat{V}_A + \beta(V_B - \hat{V}_B), \tag{6.3.12}$$

where β is a parameter with value other than 1. The new relation between the variances is now given by

$$\text{var}\left(\hat{V}_A^{\beta}\right) = \text{var}\left(\hat{V}_A\right) + \beta^2\,\text{var}\left(\hat{V}_B\right) - 2\beta\,\text{cov}\left(\hat{V}_A, \hat{V}_B\right). \tag{6.3.13}$$

The particular choice of β which minimizes var $(\hat{V}_A^{\beta})$ is found to be

$$\beta^* = \frac{\mathrm{cov}(\hat{V}_A, \hat{V}_B)}{\mathrm{var}(\hat{V}_B)}. \tag{6.3.14}$$

Unlike the choice of $\beta = 1$ used in Eq. (6.3.9), the control variate estimate based on β^* is guaranteed to decrease variance. Unfortunately, the determination of β^* requires the knowledge of $\mathrm{cov}(\hat{V}_A, \hat{V}_B)$, which is in general not available. However, one may estimate β^* using the regression technique from the simulated option values $V_A^{(i)}$ and $V_B^{(i)}$, $i = 1, 2, \cdots, M$, obtained from the simulation runs.

Valuation of Asian options

A nice example of applying the control variate method is the estimation of the value of an arithmetic averaging Asian option based on the knowledge of the exact analytic formula for the corresponding geometric averaging Asian option. The two types of Asian options are very similar in nature except that the terminal payoff function depends on either arithmetic averaging or geometric averaging of the asset price function.

The averaging feature of Asian options does not pose any difficulty in Monte Carlo simulation since the average of the asset prices at different observational instants in a given simulated path can be computed easily. Since option price formulas are readily available for the majority of geometrically averaged Asian options, the knowledge of which may be used to include a variance reduction procedure to reduce the confidence interval in the Monte Carlo simulation performed for valuation of the corresponding arithmetically averaged Asian options (Kemna and Vorst, 1990).

Let V_A denote the price of an option whose payoff depends on the arithmetic averaging of the underlying asset price and V_G be the price of an option similar to the above option except that geometric averaging is taken. How does one improve the estimation of V_A from a Monte Carlo simulation by taking advantage of the knowledge of closed form formula of V_G? Let $\hat{V}_A$ and $\hat{V}_G$ denote the discounted option payoff for a single simulated path of the asset price with respect to arithmetic and geometric averaging, respectively, so that

$$V_A = E[\widehat{V}_A] \quad \text{and} \quad V_G = E[\widehat{V}_G]. \tag{6.3.15}$$

We then have

$$V_A = V_G + E[\widehat{V}_A - \widehat{V}_G], \tag{6.3.16a}$$

and so an unbiased estimator of V_A is given by

$$\hat{V}_A^{cv} = \hat{V}_A + (V_G - \hat{V}_G). \tag{6.3.16b}$$

One then follows the variance reduction procedure that the direct estimator $\hat{V}_A$ is adjusted by the difference between the exact value V_G and the estimated value $\hat{V}_G$. The error $(V_G - \hat{V}_G)$ is employed as a control to improve the estimation of V_A.

6.3.2 Low discrepancy sequences

The crude Monte Carlo method uses random (more precisely pseudo-random) points and the rate of convergence is known to be $O\left(\frac{1}{\sqrt{M}}\right)$, where M is the number of simulation trials. The inverse square root order of convergence implies that $O\left(\frac{1}{\epsilon^2}\right)$ simulations are required to achieve $O(\epsilon)$ level of accuracy. Such a low rate of convergence is certainly not quite desirable. Also, it is quite common to have the accuracy of simulation to be sensitive to the initial seed.

It is commonly observed that the pseudo-random points may not be quite uniformly dispersed throughout the domain of the problem. It seems reasonable to postulate that convergence may be improved if these points are more uniformly distributed. A notion in number theory called *discrepancy* measures the deviation of a set of points in d dimensions from uniformity. Lower discrepancy means the points are more evenly dispersed. There have been a few well tested sequences, called quasi-random sequences (though they are deterministic in nature), which demonstrate a low level of discrepancy. Some of these examples are the Sobol points and Halton points (Paskov and Traub, 1995). These low discrepancy sequences have the nice property that when successive points are added, the entire sequence of points still remain at a similar level of discrepancy. The routines for generating these sequences are readily available in many software texts (for example, Press *et al.*, 1992).

The rate of convergence of simulation with respect to the use of different sequences can be assessed through the numerical approximation of an integral by a discrete average. If we use equally spaced points, which is simply the trapezoidal rule of numerical integration, the error is $O(M^{-2/d})$ where d is the dimension of the integral. For the Sobol points or Halton points, the rate of convergence is $O\left(\frac{(\ln M)^d}{M}\right)$. This is still in favor of $O\left(\frac{1}{\sqrt{M}}\right)$ convergence of the Monte Carlo method when d is modest.

Various numerical studies on the use of low discrepancy sequences in finance applications reveal that the errors produced are substantially lower than the corresponding errors using the crude random sequences. Paskov and Traub (1995) employed both Sobol sequences and Halton sequences to evaluate mortgage-backed security prices, which involves the evaluation of integrals with d up to 360. They showed that the Sobol sequences outperform the Halton sequences which in turn performed better than the standard Monte Carlo method. The reason for the better performance may be attributed to the smoothness of the integrand functions. Strong research interests still persist in the continual search for better low discrepancy sequences in finance applications.

6.3.3 Valuation of American options

There had been a general belief that the Monte Carlo approach can be used only for European style derivatives. The apparent difficulties of using simulation to price American options stem from the *backward* nature of the early exercise feature since there is no way of knowing whether early exercise is optimal when a particular asset value is reached at a given time. The estimated option value with respect to a given simulated path can be determined only with a pre-specified exercise policy. A variety of simulation algorithms have been proposed in the literature to tackle the above difficulties. The earliest simulation algorithm is the "bundling and sorting" algorithm proposed by Tilley (1993). The algorithm computes an estimate for the option's continuation value by using backward induction and a bundling technique. At each time instant, simulation path with similar asset prices are grouped together to obtain an estimate of the one-period-ahead option value. Another approach [Grant *et al.* (1996)] attempts to approximate the exercise boundary at each early exercise point using backward induction, then estimates the option price in a forward simulation based on the exercise policies obtained. The other approach [Broadie and Glasserman, (1997)] attempts to find efficient upper and lower bounds from simulated paths, one based on a non-recombining tree and another based on a stochastic mesh. These two high and low estimates for the option price converge asymptotically to the true option value. Rogers (2002) proposes a direct simulation approach, which is based on a dual characterization of optimal exercise policy by the holder and hedging strategy of the writer. The method involves the choice of an appropriate Lagrangian hedging martingale so that the lookback value of the excess of option exercise value over the chosen hedging strategy is minimized. The more recent and possibly most popular approach is the linear regression method via basis functions. Such algorithm involves two levels of approximation. First, the conditional expectations in the dynamic programming procedure are approximated by projections on a finite set of basis functions. Monte Carlo simulations and least squares regression techniques are used to compute the above approximated value function. Longstaff and Schwartz (2001) choose the Laguerre polynomials as the basis functions. The guidelines on the choice of the basis functions are discussed in the papers by Tsitsiklis and Van Roy (2001), Lai and Wong (2004). Clément *et al.* (2002) prove the almost sure convergence of the algorithm. Glasserman and Yu (2004) analyze the convergence of the algorithm as both the number of basis functions and the number of simulated paths increase.

Four classes of algorithms are presented below, namely, the "bundling and sorting" algorithm, method of parameterization of the early exercise boundary, stochastic mesh method and the linear regression method via basis functions. A comparison of performance of various Monte Carlo simulation approaches for pricing American style options is reported by Fu *et al.* (2001).

A comprehensive review of Monte Carlo methods in financial engineering can be found in Glasserman's text (2004).

Tilley's bundling and sorting algorithm

Tilley (1993) proposes a "bundling and sorting" algorithm which computes an estimate for the American option's continuation value using backward induction. At each time step in the simulation procedure, simulated asset price paths are ordered by asset price and bundled into groups. The method rests on the belief that the price paths within a given bundle are sufficiently alike so that they can be considered to have the same expected one-period-ahead option value. The boundary between the exercise-or-hold decisions is determined for each time step.

The options are assumed to be exercisable at specified instants $t = 1, 2, \cdots, N$. Actually, this discretization assumption transforms the American options with continuous early exercise right to the Bermudan options with discrete exercise opportunities (see Problem 6.30). The simulation procedure generates a finite sample of R asset price paths from $t = 0$ to $t = N$, where the realization of the asset price of the k^{th} price path is represented by the sequence $\{S_0(k), \cdots, S_N(k)\}$. Let d_t denote the discount factor from t to $t+1$ and D_t be the discount factor from 0 to time t, so that $D_t = d_0 d_1 \cdots d_{t-1}$. Let X be the strike price of the option. The backward induction procedure starts at $t = N - 1$. At each $t, t = 1, 2, \cdots, N - 1$, we proceed inductively according to the following steps.

1. Sort the price paths by order of asset price by partitioning the ordered paths in Q distinct bundles of P paths in each bundle ($R = QP$). We write $B_t(k)$ as the set of price paths in the bundle containing path k at time t. For each path k, compute the intrinsic value $I_t(k)$ of the option.
2. Compute the option's continuation value $H_t(k)$, defined as the present value of the expected one-period-ahead option value:

$$H_t(k) = \frac{d_t(k)}{P} \sum_{\forall j \in B_t(k)} V_{k+1}(j), \tag{6.3.17}$$

   where $V_{t+1}(j)$ has been computed in the previous time step. In particular, $V_N(j) = I_N(j)$ for all j.
3. For each path k, compare $H_t(k)$ to $I_t(k)$ and decide "tentatively" whether to exercise the option or to continue holding it. Define $x_t(k)$ as the "tentative" exercise-or-hold indicator variable, where

$$x_t(k) = \begin{cases} 1 & \text{when } I_t(k) \geq H_t(k) \\ 0 & \text{when } I_t(k) < H_t(k) \end{cases}. \tag{6.3.18a}$$

   Here, "1" and "0" represent "exercise" and "hold", respectively.
4. In general, there may be more than one bundle in which $x_t(k) = 1$ for some $k \in B_t(k)$ but 0 for other paths within the same bundle. These bundles have a "transition zone" in asset price from "hold" to "exercise" decision.

The algorithm has to be refined by creating a sharp boundary between the "hold" and "exercise" decisions. To achieve this goal, we examine the sequence $\{x_t(k) : k = 1, \cdots, R\}$, and determine the sharp boundary as the start of the first string of "1"s, the length of which exceeds the length of every subsequent string of "0"s. The path index of the leading "1" is called k_t^*. Next, we define the "update" exercise-or-hold indicator variable $y_t(k)$ by

$$y_t(k) = \begin{cases} 1 & \text{when } k \geq k_t^* \\ 0 & \text{when } k < k_t^* \end{cases}. \tag{6.3.18b}$$

5. For each path k, define the current value $V_t(k)$ of the option by

$$V_t(k) = \begin{cases} I_t(k) & \text{when } y_t(k) = 1 \\ H_t(k) & \text{when } y_t(k) = 0 \end{cases}. \tag{6.3.19}$$

The above procedure proceeds backwards from $t = N-1$ to $t = 0$. Lastly, we define the exercise-or-hold indicator variable by

$$Z_t(k) = \begin{cases} 1 & \text{if } y_t(k) = 1 \text{ and } y_s(t) = 0 \text{ for all } s < t \\ 0 & \text{otherwise} \end{cases}. \tag{6.3.20}$$

Once the exercise policy of each price path is established, the option price estimator is given by

$$\frac{1}{R}\sum_{k=1}^{R}\sum_{t=1}^{N} Z_t(k)D_t(k)I_t(k).$$

For each path k, $Z_t(k)$ equals one at only one time instant and $D_t(k)I_t(k)$ gives the discount value of the option payoff of the path.

There are several major weaknesses in Tilley's algorithm. The algorithm is not computationally efficient since it requires storage of all simulated asset price paths at all time steps. The bundling and sorting of all price paths pose stringent requirement on storage and computation even when the number of simulated paths is moderate. As shown by Tilley's own numerical experiments, there is no guarantee on the convergence of the algorithm to the true option value. Also, the extension of the algorithm to multi-asset option models can be very tedious (see Problem 6.31).

Grant-Vora-Weeks algorithm

The simulation algorithm proposed by Grant *et al.* (1996) attempts to first identify the optimal exercise price $S_{t_i}^*$ at selected instants $t_i, i = 1, 2, \cdots, N-1$ between the current t and expiration time T. The determination of the optimal exercise prices is done by simulation at successive time steps proceeding backwards in time. Once the exercise boundary is identified, the option value can be estimated by the usual simulation procedure, respecting the early exercise strategy as dictated by the known exercise boundary.

We illustrate the procedure by considering the valuation of an American put option and choosing only three time steps between the current time t

and expiration time T, where $t_0 = t$ and $t_3 = T$. Assuming a constant dividend yield q, the optimal exercise price at T is equal to $\min\left(\frac{r}{q}X, X\right)$, where X is the strike price of the option and r is the riskless interest rate. At time t_2 which is one time period prior to expiration, the put value is $X - S_{t_2}$ when $S_{t_2} \leq S^*_{t_2}$, and $E[P_T]e^{-r(T-t_2)}$ when $S_{t_2} > S^*_{t_2}$. Here, $P_T = \max(X - S_T, 0)$ denotes the put option value at expiration time T. Obviously, $E[P_T]$ is dependent on S_{t_2}. For a given value of S_{t_2}, one can perform a sufficient number of simulations to estimate $E[P_T]$. The optimal exercise price $S^*_{t_2}$ is identified by finding the appropriate value of S_{t_2} such that

$$X - S^*_{t_2} = e^{-r(T-t_2)}E[P_T|S^*_{t_2}]. \tag{6.3.21}$$

The numerical procedures try to find the simulation estimate of $e^{-r(T-t_2)}E[P_T]$ as a function of S_{t_2} by starting with S_{t_2} close to but smaller than S^*_T (since $S^*_{t_2}$ must be less that S^*_T) and repeating the simulation process for a series of S_{t_2} which decreases systematically. Once the functional dependence of the discounted expectation value $e^{-r(T-t_2)}E[P_T]$ in S_{t_2} is available, one can find a good estimate of $S^*_{t_2}$ such that Eq. (6.3.21) is satisfied.

Proceeding backwards in time, we continue to estimate the optimal exercise price at time t_1. The simulation now starts at t_1. The initial asset value S_{t_1} is first chosen with a value slightly less than $S^*_{t_2}$ and simulation is repeated with decreasing S_{t_1}. Again, we would like to find the estimate of the discounted expectation value of holding the put, and this expectation value is a function of S_{t_1}. In a typical simulation run, an asset value S_{t_2} is generated at t_2 with an initial asset value S_{t_1}. We then determine whether S_{t_2} falls in the stopping region or otherwise. If the answer is yes, the estimated put value for that simulated path is the present value of the early exercise value; otherwise, the simulation continues by generating an asset value at expiration T. The put value for this simulation path then equals the present value of the corresponding terminal payoff. This simulation procedure is repeated sufficient number of times so that an estimate of the discounted expectation value can be obtained. In a similar manner, we determine $S^*_{t_1}$ such that when S_{t_1} is chosen to be $S^*_{t_1}$, the intrinsic value $X - S^*_{t_1}$ equals the estimate of the discounted expectation value of holding the put.

Once the optimal exercise prices at t_1 and t_2 are available, one can mimic the above numerical procedure to find the estimate of the discounted expectation value of holding the put at time t_0 by performing simulation runs with an initial asset value S_{t_0}. The put value at time t_0 for a given S_{t_0} is the maximum of the estimate of the discounted expectation value obtained from simulation (taking into account the early exercise strategy as already determined at t_1 and t_2) and the intrinsic value $X - S_{t_0}$ from early exercise.

Broadie-Glasserman algorithm

The stochastic mesh algorithm of Broadie and Glasserman (1997) produces two estimators for the true option value, one biased high and the other biased low, but both asymptotically unbiased as the number of simulations tends to infinity. The two estimates together provide a conservative confidence interval for the option value.

First, a random tree with b branches per node is constructed (see Fig. 6.7 for $b = 3$) and the asset values at the nodes at time t_j are denoted by

$$S_j^{i_1 i_2 \cdots i_j}, \quad j = 1, 2, \cdots, N, \quad 1 \leq i_1, \cdots, i_j \leq b,$$

where N is the total number of time steps. The total number of nodes at time t_j will be b^j. Here, S_0 is the fixed initial state and each sequence $S_0, S_1^{i_1}, S_2^{i_1 i_2}, \cdots, S_N^{i_1 i_2 \cdots i_N}$ is a realization of the Markov process for the asset price, and two such sequences evolve independently of each other once they differ in some i_j.

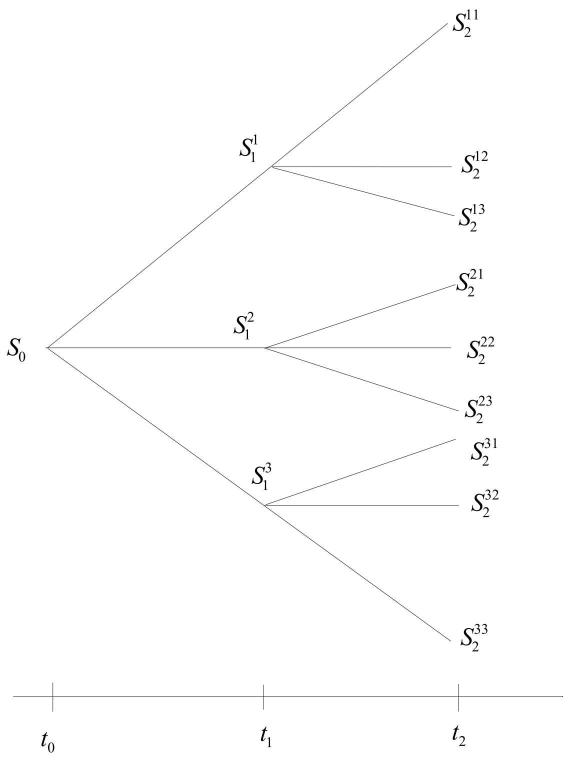

Fig. 6.7 A simulation tree with three branches and two time steps..

Let $\theta_{\text{high},j}^{i_1 \cdots i_j}$ and $\theta_{\text{low},j}^{i_1 \cdots i_j}$ denote, respectively, the high and low estimators of the option value at the $(i_1, \cdots, i_j)$th node at time t_j. Also, let $h_j(s)$ be the payoff from exercise at time t_j in state s and $1/R_{j+1}$ be the discount factor from t_j to t_{j+1}. Broadie and Glasserman defined the high estimator for the option value at the $(i_1, \cdots, i_j)$ node at time t_j to be the maximum of the early exercise payoff and the estimate of the continuation value from the b successor nodes, namely,

$$\theta_{\text{high},\ j}^{i_1 \cdots i_j} = \max\left(h_j\left(S_j^{i_1 \cdots i_j}\right), \frac{1}{b}\sum_{i_{j+1}=1}^{b}\frac{1}{R_{j+1}}\theta_{\text{high},\ j+1}^{i_1 \cdots i_j i_{j+1}}\right). \quad (6.3.22)$$

Simple arguments can be used to explain why the above estimate is biased high. If the asset prices at the nodes at time t_{j+1} turn out to be too high in the simulation process, the above dynamic programming procedure will choose not to exercise and take a value higher than the optimal decision to exercise. On the other hand, if the simulated asset prices at t_{j+1} turn out to be too low, the dynamic programming procedure will choose to exercise even when the optimal decision is not to exercise. The option value is over-estimated since we have taken advantage of knowledge of the future.

The numerical algorithm for the low estimator is slightly more complicated. At each node, one branch is used to estimate the continuation value and the other $b-1$ branches are used to estimate the exercise decision. The same procedure is repeated b times, where each branch is chosen in turn. To explain the procedure in more detail, suppose the k^{th} branch is chosen to estimate the continuation value while the other $b-1$ branches are used to estimate the exercise decision. Early exercise is chosen if the payoff $h_j\left(S_j^{i_1 \cdots i_j}\right)$ is greater than or equal to the expectation of the continuation value. This expectation is computed by taking the average among $b-1$ branches of the discounted values $\dfrac{1}{R_{j+1}}\theta_{\text{low},j+1}^{i_1 \cdots i_{j+1}}, i_{j+1} = 1, \cdots, b, i_{j+1} \neq k$. If early exercise is chosen, then the estimate $\eta_j^{i_1 \cdots i_j k}$ takes the payoff value $h_j(S_j^{i_1 \cdots i_j})$, otherwise, it takes the continuation value $\dfrac{1}{R_{j+1}}\theta_{\text{low},j+1}^{i_1 \cdots i_j k}$. Thus b estimates are obtained in these b steps of calculations and they are then averaged to determine the option value estimate at the node. The procedure can be succinctly described as follows. Write

$$\eta_j^{i_1 \cdots i_j k} = \begin{cases} h_j\left(S_j^{i_1 \cdots i_j}\right), & \text{if } h_j\left(S_j^{i_1 \cdots i_j}\right) \geq \dfrac{1}{b-1}\displaystyle\sum_{\substack{i_{j+1}=1 \\ i_{j+1}\neq k}}^{b}\dfrac{1}{R_{j+1}}\theta_{\text{low},\ j+1}^{i_1 \cdots i_j i_{j+1}} \\ \frac{1}{R_{j+1}}\theta_{\text{low},\ j+1}^{i_1 \cdots i_j k}, & \text{if } h_j\left(S_j^{i_1 \cdots i_j}\right) < \dfrac{1}{b-1}\displaystyle\sum_{\substack{i_{j+1}=1 \\ i_{j+1}\neq k}}^{b}\dfrac{1}{R_{j+1}}\theta_{\text{low},\ j+1}^{i_1 \cdots i_j i_{j+1}}, \end{cases}$$

$$k = 1, \cdots, b, \quad (6.3.23a)$$

then

$$\theta_{\text{low},\ j}^{i_1\cdots i_j} = \frac{1}{b}\sum_{k=1}^{b}\eta_j^{i_1\cdots i_j k}. \tag{6.3.23b}$$

The explanation why the above procedure gives a biased low estimator is relegated to an exercise (see Problem 6.35).

Both algorithms (6.3.22) and (6.3.23a,b) are backward induction, that is, knowing estimates at time t_{j+1}, we compute estimates at t_j one period earlier. For both high and low biased estimators, the starting iterates at expiration time $T = t_N$ are both given by the following terminal payoff function

$$\theta_N^{i_1\cdots i_N} = h_N\left(S_N^{i_1\cdots i_N}\right). \tag{6.3.24}$$

The Broadie-Glasserman algorithm can be extended to deal with multi-asset options, and the computation can be made parallelized to work on a cluster of workstations. Variance reduction techniques can also be employed to fasten the rate of convergence. The algorithm can allow multiple decisions other than the two-fold decision: exercise or hold.

Linear regression method via basis functions

Under the discrete assumption of exercise opportunities, the option values satisfy the following dynamic programming equations

$$V_n = \max(h_n(S), H_n(S)), \quad n = 0, 1, \cdots, N-1, \tag{6.3.25}$$

where $H_n(S)$ is the continuation value at time $t_n, S(t_n) = S, h_n(S)$ is the exercise payoff. At maturity date $t_N = T$, we have $V_N(S) = h_N(S)$ [for notational convenience, we set $H_N(S) = 0$]. The continuation values at different time instants are given by the following recursive scheme

$$H_n(S) = E\left[\max(h_{n+1}(S(t_{n+1})), H_{n+1}(S(t_{n+1}))|S(t_n) = S\right]. \tag{6.3.26}$$

The difficulty of estimating the above conditional expectations may be resolved by considering an approximation of the form

$$H_n(S) \approx \sum_{m=0}^{M}\alpha_{nm}\phi_{nm}(S), \tag{6.3.27}$$

for some choice of basis functions $\phi_{nm}(S)$. Longstaff and Schwartz (2001) propose to determine the coefficients α_{nm} through least squares projection onto the span of basis functions. Their chosen basis functions are the Laguerre polynomials defined by

$$L_m(S) = e^{-S/2}\frac{e^S}{m!}\frac{d^m}{dS^m}\left(S^m e^{-S}\right), \quad m = 0, 1, 2, \cdots. \tag{6.3.28}$$

The first few members are $L_0(S) = e^{-S/2}, L_1(S) = e^{-S/2}(1-S), L_2(S) = e^{-S/2}\left(1 - 2S + \dfrac{S^2}{2}\right)$.

Following the description of the algorithm by Longstaff and Schwartz (2001), we use $C(\omega, s; t, T)$ to denote the path of cash flows generated by the option, conditional on the option not being exercised at or prior to time t. Here, ω represents a sample path and T is option's maturity date. The holder is assumed to follow the optimal stopping strategy for all subsequent times s, where $t < s \leq T$. Recall that the value of an American option is given by maximizing the discounted cash flows from the option, where the maximum is taken over all stopping times. We seek for a pathwise approximation to the optimal stopping rule associated with the early exercise right in the American option. Like other simulation algorithms, the key is to identity the conditional expected value of continuation.

Let $H_n(\omega; t_n)$ denote the continuation value at time t_n. By no arbitrage principle, $H_n(\omega)$ is given by the expectation of the remaining discounted cash flows under the risk neutral measure. At time $t_n, H_n(\omega)$ is given by

$$H_n(\omega; t_n) = E\left[\sum_{j=n+1}^{N} e^{-r(t_j - t_n)} C(\omega, t_j; t_n, T)\right], \tag{6.3.29}$$

where the expectation is taken under the risk neutral measure conditional on the filtration at time t_n. Suppose we have chosen M basis functions, then $H_n(\omega)$ is estimated by regressing the discounted cash flow onto the basis functions for the paths where the option is in-the-money at time t_n. Longstaff and Schwartz propose that only in-the-money paths are used in the estimation since the exercise decision is relevant only in the in-the-money regime. Once the functional form of the estimated continuation value $\widehat{H}_n(\omega)$ is obtained from linear regression, we can calculate the estimated continuation value from the known asset price at time t_n for that path ω.

Our goal is to solve for the stopping rule that maximizes the option value at every time point along each asset price path. We start from the maturity date t_N, and proceed backwards in time. At t_N, the cash flows are given by the terminal payoff function and thus they are readily known. At one time step backward, we search for those paths that are in-the-meney at t_{N-1}. From these paths, we compute the discounted cash flow received at time t_N given that the option remains alive at time t_{N-1}. Consider path k, its asset price at t_{N-1} and t_N are denoted by $S_{N-1}^{(k)}$ and $S_N^{(k)}$, respectively, $k = 1, \cdots, K$, where K is the total number of paths that are in-the-money at t_{N-1}. The discounted cash flow at t_{N-1} for path k is given by $e^{-r(t_N - t_{N-1})} h_N(S_N^{(k)})$, where h_N is the terminal payoff function of the option. Using the information of these K data points and choosing M basis functions, we estimate the continuation value $\widehat{H}_{N-1}^{(k)}$ by regressing the discounted cash flow at t_{N-1} with respect to the asset price at t_{N-1}. Early exercise at time t_{N-1} is optimal for an in-the-money path ω if the immediate exercise value is greater than or equal to the estimated continuation value. In this case, the cash flow at t_{N-1} is set equal the exercise value. A numerical example illustrating the details of this regression procedure can be found in Longstaff-Schwartz's paper (2001).

Once the cash flow paths and stopping rule at t_{N-1} have been determined, we then proceed recursively in the same manner to the earlier time points $t_{N-2}, \cdots, t_1$. As a result, we obtain the optimal stopping rule at all time points for every path. Once the cash flows generated by the option for all paths are identified, we can compute an estimate of the option value by discounting each cash flow back to the issue date and averaging over all sample price paths.

6.4 Problems

6.1 Instead of the tree-symmetry condition: $u = 1/d$ [see Eq. (6.1.1c)], Jarrow and Rudd (1983) choose the third condition to be $p = 1/2$. By solving the above condition together with Eqs. (6.1.1a,b), show that

$$u = R(1 + \sqrt{e^{\sigma^2 \Delta t} - 1}), d = R(1 - \sqrt{e^{\sigma^2 \Delta t} - 1}) \quad \text{and} \quad p = \frac{1}{2}.$$

6.2 Suppose the underlying asset is paying a continuous dividend yield at the rate q, the two governing equations for u, d and p are modified as

$$pu + (1-p)d = e^{(r-q)\Delta t}$$
$$pu^2 + (1-p)d^2 = e^{2(r-q)\Delta t} e^{\sigma^2 \Delta t}.$$

Show that the parameter values in the binomial model are modified by replacing the growth factor of the asset price under the risk neutral measure $e^{r\Delta t}$ by the new factor $e^{(r-q)\Delta t}$ while the discount factor in the binomial formula remains to be $e^{-r\Delta t}$.

6.3 Show that

$$\lim_{n \to \infty} \Phi(n, k, p') = N(d_1)$$

where $p' = ue^{-r\Delta t}p$ and $d_1 = \dfrac{\ln \frac{S}{X} + \left(r + \frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}$.

Hint: Note that

$$1 - \Phi(n, j, p') = P\left[\frac{j - np'}{\sqrt{np'(1-p')}} < \frac{\ln \frac{X}{S} - n\left(p' \ln \frac{u}{d} + \ln d\right) - \alpha \ln \frac{u}{d}}{\sqrt{np'(1-p')} \ln \frac{u}{d}}\right],$$
$$0 < \alpha \leq 1.$$

By considering the Taylor expansion of $n\left(p'\ln\frac{u}{d}+\ln d\right)$ and $np'(1-p')\left(\ln\frac{u}{d}\right)^2$ in power of Δt, show that

$$\lim_{n\to\infty} n\left(p'\ln\frac{u}{d}+\ln d\right) = \left(r+\frac{\sigma^2}{2}\right)\tau$$
$$\lim_{n\to\infty} np'(1-p')\left(\ln\frac{u}{d}\right)^2 = \sigma^2\tau,$$

where $n\Delta t = \tau$.

6.4 Consider the modified binomial formula employed for the numerical valuation of an American put on a non-dividend paying asset [see Eq. (6.1.26)], deduce the optimal exercise price at time close to expiry from the binomial formula. Compare the result with that of the continuous model by taking the limit $\Delta t \to 0$.

6.5 Consider the nodes in the binomial tree employed for the numerical valuation of an American put option on a non-dividend paying asset. The $(n,j)^{\text{th}}$ node corresponds to the node which is n time steps from the current time and has j upward moves. The put value at the $(n,j)^{\text{th}}$ node is denoted by P_j^n. Similar to the continuous models, we define the stopping region $\mathcal{S}$ and continuation region $\mathcal{C}$ by

$$\mathcal{S} = \left\{(n,j)|P_j^n = X - Su^jd^{n-j}\right\}$$
$$\mathcal{C} = \left\{(n,j)|P_j^n > X - Su^jd^{n-j}\right\},$$

that is $\mathcal{S}\,(\mathcal{C})$ represents the set of nodes where the put is dead (alive). Let N be the total number of time steps in the tree. Prove the following properties of $\mathcal{S}$ and $\mathcal{C}$ (Kim and Byun, 1994):

(i) Suppose both $(n+1,j)$ and $(n+1,j+1)$ belong to $\mathcal{S}$, then $(n,j)\in\mathcal{S}$ for $0\le n\le N-1, 0\le j\le n$.

(ii) Suppose $(n+2,j+1)\in\mathcal{C}$, then $(n,j)\in\mathcal{C}$ for $0\le n\le N-2, 0\le j\le n$.

(iii) Suppose $(n,j)\in\mathcal{S}$, then both $(n,j-1)$ and $(n-1,j-1)\in\mathcal{S}$; also, suppose $(n,j)\in\mathcal{C}$, then $(n,j+1)\in\mathcal{C}$ and $(n-1,j)\in\mathcal{C}$, for $1\le n\le N, 1\le j\le n-1$.

6.6 Consider the pricing of the callable American call option by binomial calculations, let us write

$$C_{cont} = \frac{pC_{j+1}^{n+1}+(1-p)C_j^{n+1}}{R}.$$

In the continuation region, we must have $S\le K+X$. Show that binomial scheme (6.1.27) can be simplified to become

$$C_j^n = \min(K, \max(C_{cont}, S_j^n - X)).$$

6.7 Another possible binomial algorithm for pricing the callable American call option can be constructed as follow

$$C_j^n = \max(\min(C_{cont}, K), S_j^n - X).$$

The added procedure $\min(C_{cont}, K)$ compares C_{cont} and K to test whether the position on the issuer can be improved by calling the option. Show that the above scheme is equivalent to binomial scheme (6.1.27).

6.8 Show that the total number of multiplications and additions in performing n steps of numerical calculations using the trinomial and binomial schemes are given by

Scheme	*Number of multiplications*	*Number of additions*
trinomial	$3n^2$	$2n^2$
binomial	$n^2 + n$	$\frac{1}{2}(n^2 + n)$

6.9 Suppose we let $p_2 = 0$ and write $p_1 = -p_3 = p$ in the trinomial scheme. By matching the mean and variance of $\zeta(t)$ and $\zeta^a(t)$ accordingly

$$E[\zeta^a(t)] = 2pv - v = \left(r - \frac{\sigma^2}{2}\right) \triangle t$$
$$\mathrm{var}(\zeta^a(t)) = v^2 - E[\zeta^a(t)]^2 = \sigma^2 \triangle t,$$

show that the parameters v and p obtained by solving the above pair of equations are found to be

$$v = \sqrt{\left(r - \frac{\sigma^2}{2}\right)^2 \triangle t^2 + \sigma^2 \triangle t}$$
$$p = \frac{1}{2}\left[1 + \frac{\left(r - \frac{\sigma^2}{2}\right)\triangle t}{\sqrt{\sigma^2 \triangle t + \left(r - \frac{\sigma^2}{2}\right)^2 \triangle t^2}}\right].$$

6.10 Boyle (1988) proposes the following three-jump process for the approximation of the asset price movement over one period:

nature of jump	*probability*	*asset price*
up	p_1	uS
horizontal	p_2	S
down	p_3	dS

where S is the current asset price. The middle jump ratio m is chosen to be 1. There are five parameters in Boyle's trinomial model. The governing equations for the parameters can be obtained by

(i) setting sum of probabilities to be 1

$$p_1 + p_2 + p_3 = 1,$$

(ii) equating the first two moments of the approximating discrete distribution and the corresponding continuous lognormal distribution of the Black-Scholes model

$$p_1 u + p_2 + p_3 d = e^{r\triangle t} = R$$

$$p_1 u^2 + p_2 + p_3 d^2 - (p_1 u + p_2 + p_3 d)^2 = e^{2r\triangle t}\ (e^{\sigma^2 \triangle t} - 1).$$

The last equation can be simplified as

$$p_1 u^2 + p_2 + p_3 d^2 = e^{2r\triangle t}\ e^{\sigma^2 \triangle t}.$$

The remaining two conditions can be chosen freely. They are chosen by Boyle to be

$$ud = 1$$

and

$$u = e^{\lambda\sigma\sqrt{\triangle t}}, \qquad \lambda \text{ is a free parameter.}$$

By solving the five equations together, show that p_1 and p_3:

$$p_1 = \frac{(W-R)u - (R-1)}{(u-1)(u^2-1)}, \qquad p_3 = \frac{(W-R)u^2 - (R-1)u^3}{(u-1)(u^2-1)},$$

where $W = R^2 e^{\sigma^2 \Delta t}$. Also show that Boyle's trinomial model reduces to the Cox-Ross-Rubinstein binomial scheme when $\lambda = 1$.

6.11 Suppose we let $y = \ln S$, the Kamrad-Ritchken trinomial scheme can be expressed as

$$c(y, t - \triangle t) = [p_1 c(y+v, t) + p_2 c(y, t) + p_3(y - v, t)]\ e^{-r\triangle t}.$$

Show that the Taylor expansion of the above trinomial scheme is given by

$$\begin{aligned} &- c(y, t - \triangle t) + [p_1 c(y+v, t) + p_2 c(y, t) + p_3(y - v, t)]\ e^{-r\triangle t} \\ &= \triangle t \frac{\partial c}{\partial t}(y, t) - \frac{\Delta t^2}{2}\frac{\partial^2 c}{\partial t^2}(y, t) + \cdots + (1 - e^{-r\triangle t}) c(y, t) \\ &\quad + e^{-r\triangle t}\left[(p_1 - p_3) v \frac{\partial c}{\partial y} + \frac{1}{2}(p_1 + p_3) v^2 \frac{\partial^2 c}{\partial y^2} \right. \\ &\qquad \left. + \frac{1}{6}(p_1 - p_3) v^3 \frac{\partial^3 c}{\partial y^3} + \cdots \right]. \end{aligned}$$

Given the probability values in Eqs. (6.1.33a,b,c), show that $c(y, t)$ satisfies

$$0 = \frac{\partial c}{\partial t}(y,t) + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial c}{\partial y}(y,t) + \frac{\sigma^2}{2}\frac{\partial^2 c}{\partial y^2}(y,t) - rc(y,t) + O(\triangle t).$$

6.12 Show that the width of the domain of dependence of the trinomial scheme (see Figure 6.5) increases as $\sqrt{n}$, where n is the number of time steps to expiry.

6.13 Consider the 5-point multinomial scheme defined in Eq. (6.1.37a-e) and the corresponding 4-point scheme (obtained by setting $\lambda = 1$), show that the total number of multiplications and additions in performing n steps of the schemes are given by (Kamrad and Ritchken, 1991)

Scheme	*Number of multiplications*	*Number of additions*
5-point	$\frac{5}{3}(2n^3 + n)$	$\frac{4}{3}(2n^3 + n)$
4-point	$\frac{2}{3}(2n^3 + 3n^2 + n)$	$\frac{1}{2}(2n^3 + 3n^2 + n)$

6.14 Consider a three-state option model where the logarithmic return processes of the underlying assets are given by

$$\ln \frac{S_i^{\Delta t}}{S_i} = \zeta_i, \quad i = 1, 2, 3.$$

Here, ζ_i denotes a normal random variable with mean $\left(r - \frac{\sigma_i^2}{2}\right)\Delta t$ and variance $\sigma_i^2 \Delta t, i = 1, 2, 3$. Let ρ_{ij} denote the instantaneous correlation coefficient between ζ_i and $\zeta_j, i, j = 1, 2, 3, i \neq j$. Suppose the approximating multi-variate distribution $\xi_i^a, i = 1, 2, 3$, is taken to be

ζ_1^a	ζ_2^a	ζ_3^a	probability
v_1	v_2	v_3	p_1
v_1	v_2	$-v_3$	p_2
v_1	$-v_2$	v_3	p_3
v_1	$-v_2$	$-v_3$	p_4
$-v_1$	v_2	v_3	p_5
$-v_1$	v_2	$-v_3$	p_6
$-v_1$	$-v_2$	v_3	p_7
$-v_1$	$-v_2$	$-v_3$	p_8
0	0	0	p_9

where $v_i = \lambda\sigma_i\sqrt{\Delta t}, i = 1, 2, 3$. Following the Kamrad-Ritchken approach, find the probability values so that the approximating discrete distribution converges to the continuous multi-variate distribution as $\Delta t \to 0$.

Hint: The first and the last probability values are given by

$$p_1 = \frac{1}{8}\left\{\frac{1}{\lambda^2} + \frac{\sqrt{\Delta t}}{\lambda}\left(\frac{r - \frac{\sigma_1^2}{2}}{\sigma_1} + \frac{r - \frac{\sigma_2^2}{2}}{\sigma_2} + \frac{r - \frac{\sigma_3^2}{2}}{\sigma_3}\right),\right.$$
$$\left. + \frac{\rho_{12} + \rho_{13} + \rho_{23}}{\lambda^2}\right\}$$
$$p_9 = 1 - \frac{1}{\lambda^2}.$$

6.15 The consecutive Parisian feature counts the number of consecutive breaching occurences that the asset price process stays in the knock-out region. The count is reset to zero once the asset price moves out from the knock-out region. Let $V_{j,k}^n$ denote the option value of a consecutive Parisian option at the $(n, j)^{th}$ node on a trinomial tree, where the index k counts the number of consecutive breaching occurrences. Construct the corresponding forward shooting grid algorithm for pricing options with the consecutive Parisian feature (Kwok and Lau, 2001a).

6.16 Consider the window Parisian feature, a moving window is defined with $\widehat{m}$ consecutive monitoring instants at or before the current time. The option is knocked out when the asset price falls within the knock-out region exactly m times, $m \leq \widehat{m}$, within the window. Under what condition does the window Parisian feature reduce to the consecutive Parisian feature? How to construct the corresponding discrete grid function g_{win}?

Hint: We define a binary string $A = a_1a_2 \cdots a_{\widehat{m}}$ to represent the history of asset price path falling within or outside the knock-out region within the window. For the window Parisian feature, the associated path dependence state vector has binary strings as elements (Kwok and Lau, 2001a).

6.17 Construct the FSG scheme for pricing the continuously monitored European style floating strike lookback call option. In particular, describe how to define the terminal payoff values. How to modify the FSG scheme in order to incorporate the American early exercise feature?

6.18 Consider the European put option with the automatic strike reset feature, where the strike price is reset to the prevailing asset price on a pre-specified reset date if the option is out-of-the-money on that date. The strike price is not known aprior, rather it depends on the actual realization of the asset price on those prespecified reset dates. Construct the FSG scheme that prices the strike reset put option (Kwok and Lau, 2001a).

Hint: Let $t_\ell, \ell = 1, 2, \cdots, m$ be the prespecified reset dates, and let X_ℓ denote the strike price reset at t_ℓ. Explain why

$$X_\ell = \max(X, X_{\ell-1}, S(t_\ell)),$$

where X is the original strike price.

6.19 Suppose we would like to approximate $\dfrac{df}{dx}$ at x_0 up to $O(\Delta x^2)$ using function values at $x_0, x_0 - \Delta x$ and $x - 2\Delta x$, that is,

$$\left.\frac{df}{dx}\right|_{x_0} = \alpha_{-2} f(x_0 - 2\Delta x) + \alpha_{-1} f(x_0 - \Delta x) + \alpha_0 f(x_0) + O(\Delta x^2),$$

where α_{-2}, α_{-1} and α_0 are unknown coefficients to be determined. Show that these coefficients are obtained by solving

$$\begin{pmatrix} 1 & 1 & 1 \\ -2 & -1 & 0 \\ 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_{-2} \\ \alpha_{-1} \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

6.20 Consider the following difference operators, show that they approximate the corresponding differential operator up to second order accuracy

(i) $$\left.\frac{d^2 f}{dx^2}\right|_{x_0} = \frac{2f(x_0) - 5f(x_0 - \Delta x) + 4f(x - 2\Delta x) - f(x_0 - 3\Delta x)}{\Delta x^2} + O(\Delta x^2)$$

(ii) $$\frac{\partial^2 f}{\partial x \partial y} = [f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0 + \Delta x, y_0 - \Delta y) - f(x_0 - \Delta x), y_0 + \Delta y) + f(x_0 - \Delta x, y_0 - \Delta y)]/(4\Delta x \Delta y) + O(\Delta x^2) + O(\Delta y^2).$$

6.21 Show that the leading truncation error terms of the Crank-Nicolson scheme

$$\begin{aligned} \frac{V_j^{n+1} - V_j^n}{\Delta\tau} &= \frac{\sigma^2}{4}\left(\frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\Delta x^2} + \frac{V_{j+1}^{n+1} - 2V_j^{n+1} + V_{j-1}^{n+1}}{\Delta x^2}\right) \\ &\quad + \frac{1}{2}\left(r - \frac{\sigma^2}{2}\right)\left(\frac{V_{j+1}^n - V_{j-1}^n}{2\Delta x} + \frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2\Delta x}\right) \\ &\quad - \frac{r}{2}(V_j^n + V_j^{n+1}) \end{aligned}$$

are $O(\Delta\tau^2, \Delta x^2)$.

Hint: Perform the Taylor expansion at $\left(j\Delta x, \left(n + \frac{1}{2}\right)\Delta\tau\right)$.

6.22 Consider the following form of the Black-Scholes equation:

$$\frac{\partial W}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial x^2} + \left(r - q - \frac{\sigma^2}{2}\right)\frac{\partial W}{\partial x}, \quad W = e^{-r\tau}V \text{ and } x = \ln S,$$

where $V(S,\tau)$ is the option price and S is the asset price. The two-level six-point implicit *compact scheme* is given by

$$a_1 W_{j+1}^{n+1} + a_0 W_j^{n+1} + a_{-1} W_{j-1}^{n+1} = b_1 W_{j+1}^n + b_0 W_j^n + b_{-1} W_{j-1}^n$$

where

$$c = \left(r - q - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{\Delta x}, \quad \mu = \sigma^2 \frac{\Delta\tau}{\Delta x^2},$$
$$a_1 = 1 - 3\mu - 3c - \frac{c^2}{\mu} + \frac{c}{\mu}, \quad a_0 = 10 + 6\mu + \frac{2c^2}{\mu},$$
$$a_{-1} = 1 - 3\mu + 3c - \frac{c^2}{\mu} - \frac{c}{\mu}, \quad b_1 = 1 + 3\mu + 3c + \frac{c^2}{\mu} + \frac{c}{\mu},$$
$$b_0 = 10 - 6\mu - \frac{2c^2}{\mu}, \quad b_{-1} = 1 + 3\mu - 3c + \frac{c^2}{\mu} - \frac{c}{\mu}.$$

Show that the compact scheme is second order time accurate and fourth order space accurate.

6.23 Use the Fourier method to deduce the von Neumann stability condition for (i) Jarrow-Rudd binomial scheme (see Problem 6.1), (ii) Kamrad-Ritchken trinomial scheme, (iii) explicit FTCS scheme.

6.24 Let $p(S, M, t)$ denote the price function of the floating strike lookback put option. Define $x = \ln \frac{M}{S}$ and $V(x,t) = \frac{p(S,M,t)}{S}$. The pricing formulation of $V(x,t)$ is given by

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}\frac{\partial^2 V}{\partial x^2} + \left(q - r - \frac{\sigma^2}{2}\right)\frac{\partial V}{\partial x} - qV = 0, \quad x > 0, 0 < t < T.$$

The final and boundary conditions are

$$V(x,T) = e^x - 1 \quad \text{and} \quad \frac{\partial V}{\partial x}(0,t) = 0,$$

respectively. By writing $\alpha = \frac{1}{2} + \frac{\Delta x}{2}\left(\frac{r-q}{\sigma^2} + \frac{1}{2}\right)$ and setting $\Delta x = \sigma\sqrt{\Delta t}$, the binomial scheme takes the form

$$V_j^n = \frac{1}{1 + q\Delta t}\left[\alpha V_{j-1}^{n+1} + (1-\alpha)V_{j+1}^{n+1}\right], \quad j \geq 0.$$

Suppose the boundary condition at $x = 0$ is approximated by

$$V_{-1}^{n+1} = V_0^{n+1},$$

then the numerical boundary value is given by

$$V_0^n = \frac{1}{1+q\Delta t}\left[\alpha V_0^{n+1} + (1-\alpha)V_1^n\right].$$

Let $\mathcal{T}_0^n$ denote the truncation error at $j = 0$ of the above binomial scheme, show that (Dai, 2001)

$$\mathcal{T}_0^n = -\frac{1}{1+q\Delta t}\frac{\sigma^2}{4}\frac{\partial^2 V}{\partial x^2}\bigg|_{x=0} + O(\Delta x).$$

Therefore, the proposed binomial scheme is not consistent.

6.25 To obtain a consistent binomial scheme for the floating strike lookback put option, we derive the binomial discretization at $j = 0$ using the finite volume approach (Dai, 2001). First, we integrate the governing differential equation from $x = 0$ to $x = \frac{\Delta x}{2}$ to obtain

$$\begin{aligned} 0 = &\int_0^{\frac{\Delta x}{2}} \left(\frac{\partial V}{\partial t} - qV\right) dx + \frac{\sigma^2}{2}\left[\frac{\partial V}{\partial x}\bigg|_{\frac{\Delta x}{2}} - \frac{\partial V}{\partial x}\bigg|_0\right] \\ &+ \left(q - r - \frac{\sigma^2}{2}\right)\left(V_{\frac{\Delta x}{2}} - V_0\right). \end{aligned}$$

Suppose we adopt the following approximations:

$$\int_0^{\frac{\Delta x}{2}} \left(\frac{\partial V}{\partial t} - qV\right) dx \approx \left(\frac{V_0^{n+1} - V_0^n}{\Delta t} - qV_0^n\right)\Delta x$$

$$\frac{\partial V}{\partial x}\bigg|_{\frac{\Delta x}{2}} \approx \frac{V_1^{n+1} - V_0^{n+1}}{\Delta x}, \quad V_{\frac{\Delta x}{2}} \approx \frac{V_1^{n+1} + V_0^{n+1}}{2},$$

show that the binomial approximation at $j = 0$ is given by

$$V_0^n = \frac{1}{1+q\Delta t}\left[(2\alpha - 1)V_0^{n+1} + 2(1-\alpha)V_1^{n+1}\right].$$

Examine the consistency of the above binomial approximation.

6.26 Suppose we use the FTCS scheme to solve the Black-Scholes equation so that

$$\frac{V_j^{n+1} - V_j^n}{\Delta\tau} = \frac{\sigma^2}{2}S_j^2\frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\Delta S^2} + rS_j\frac{V_{j+1}^n - V_{j-1}^n}{2\Delta S} - rV_j^n.$$

Show that the sufficient conditions for non-appearance of spurious oscillations in the numerical scheme are given by (Zvan *et al.*, 1998)

$$\Delta S < \frac{\sigma^2 S_i}{r} \quad \text{and} \quad \frac{1}{\Delta\tau} > \frac{\sigma^2 S_i^2}{\Delta S^2} + r.$$

6.27 A sequential barrier option has two-sided barriers. Unlike the usual double barrier options, the order of breaching of the barrier is specified. The second barrier is activated only after the first barrier has been hit earlier, and the option is knocked out only if both barriers have been hit in the pre-specified order. Construct the explicit finite difference scheme for pricing this sequential barrier option under the Black-Scholes pricing framework (Kwok *et al.*, 2001).

6.28 The penalty method is characterized by the replacement of the linear complementarity formulation of the American option by appending a non-linear penalty term in the Black-Scholes equation. Let $V^*(S,\tau)$ denote the exercise payoff of an American option. The non-linear penalty term takes the form $\rho \max(V^* - V, 0)$, where ρ is the positive penalty parameter and $V(S,\tau)$ is the option price function. It can be shown that when $\rho \to \infty$, the solution of the following equation

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV + \rho \max(V^* - V, 0)$$

gives the solution of the American option price function. Discuss the construction of the Crank-Nicolson scheme for solving the above non-linear differential equation, paying special attention to the solution of the resulting non-linear algebraic system of equations. Note that the non-linearity stems from the penalty term (Forsyth and Vetzal, 2002).

6.29 Consider the antithetic variates method [see Eqs. (6.3.7a,b)], explain why

$$\text{var}\left(\frac{c_i + \widetilde{c}_i}{2}\right) = \frac{1}{2}\left[\text{var}(c_i) + \ \text{cov}(c_i, \widetilde{c}_i)\right].$$

Note that the amount of computational work to generate $\bar{c}_{AV}$ [see Eq. (6.3.8)] is about twice the work to generate $\hat{c}$. By applying inequality (6.3.5), show that the antithetic variates method improves efficiency provided that

$$\text{cov}(c_i, \widetilde{c}_i) \le 0.$$

Give a statistical justification why the above negative correlation property is in general valid (Boyle *et al.*, 1997).

6.30 Consider the Bermudan option pricing problem, where the Bermudan option has d exercise opportunities at times $t_1 < t_2 < \cdots < t_d = T$, with $t_1 \ge 0$. Here, the issue date and maturity date of the Bermudan option are taken to be 0 and T, respectively. Let M_t denote the value at time t of \$1 invested in the riskless money market account at time 0. Let h_t denote the payoff from exercise at time t and τ^* be a stopping time taking values in $\{t_1, t_2, \cdots, t_d\}$. The value of the Bermudan option at time 0 is given by

$$V_0 = \sup_{\tau^*} E_0 \left[\frac{h_\tau}{M_\tau} \right].$$

Consider the quantity defined by (Andersen and Broadie, 2004)

$$Q_{t_i} = \max \left(h_{t_i}, E_{t_i} \left[\frac{M_{t_i}}{M_{t_{i+1}}} Q_{t_{i+1}} \right] \right), \quad i = 1, 2, \cdots, d-1,$$

explain why Q_{t_i} gives the value of a Bermudan option newly issued at time t_i. Is it the same as the value at t_i of a Bermudan option issued at time 0? If not, explain why?

6.31 It has been generally believed that the extension of the Tilley algorithm to multi-asset American options is not straightforward. Discuss the modifications on the bundling and sorting procedure required in the path grouping of all the asset price paths of the n assets, $n > 1$. Also, think about how to determine the exercise-or-hold indicator variables when the exercise boundary is defined by a high-dimensional surface (Fu *et al.*, 2001).

6.32 Discuss how to implement the secant method in the root-finding procedure of solving the optimal exercise price $S^*_{t_i}$ from the following algebraic equation

$$X - S^*_{t_i} = e^{-r(T-t_i)} E \left[P_{i+1} \middle| S_{t_i} = S^*_{t_i} \right]$$

in the Grant-Vora-Weeks algorithm (Fu *et al.*, 2001).

6.33 Judge whether the simulation estimator on the option price given by the Grant-Vora-Weeks algorithm is biased high or low or unbiased.

6.34 Explain why the estimator $\theta^{i_1 \cdots i_j}_{\text{low},j}$ defined by Eqs. (6.3.23a,b) is biased low.

Hint: Upward bias is eliminated since the continuation value and the early exercise decision are determined from independent information sets. The early exercise decision is always suboptimal with a finite sample.