MAFS5250 – Computational Methods for Pricing Structured Products

Topic 3 – Finite difference methods

3.1 Discretization of the Black-Scholes equation

- Explicit schemes
  - Domain of dependence
  - Incorporation of boundary conditions
  - Skew computational stencil
- Crank-Nicolson scheme
  - Thomas algorithm
- Iterated method for pricing American options
- Two-asset models

3.2 Numerical approximation of auxiliary conditions

- Initial conditions and terminal payoffs
- Lookback options - Neumann boundary condition
- Discretely monitored lookback and barrier options
- Down-barrier proportional step call option

3.3 Properties of numerical solutions

- Truncation errors and order of convergence
- Numerical stability
- Spurious oscillations

3.1 Discretization of the Black-Scholes equation

Black-Scholes equation: $\dfrac{\partial V}{\partial t} + rS\dfrac{\partial V}{\partial S} + \dfrac{\sigma^2}{2}S^2\dfrac{\partial^2 V}{\partial S^2} - rV = 0.$

Use the transformed variables: $\tau = T - t, x = \ln S$,

$$\begin{aligned}\frac{\partial}{\partial t} &= -\frac{\partial}{\partial \tau}, \quad \frac{\partial}{\partial S} = \frac{1}{S}\frac{\partial}{\partial x} \quad \text{or} \quad S\frac{\partial}{\partial S} = \frac{\partial}{\partial x}\\ \frac{\partial^2}{\partial x^2} &= S\frac{\partial}{\partial S}\left(S\frac{\partial}{\partial S}\right) = S^2\frac{\partial^2}{\partial S^2} + S\frac{\partial}{\partial S} \text{ so that } S^2\frac{\partial^2}{\partial S^2} = \frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial x}.\end{aligned}$$

The transformed Black-Scholes equation now has constant coefficients:

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

To absorb the discount term, we let $W = e^{r\tau}V$, then

$$\frac{\partial W}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial W}{\partial x}, \quad \tau > 0, -\infty < x < \infty.$$

Remark

There has always been a debate on the choice of either S or $x = \ln S$ as the independent state variable.

- If S is used, then the diffusion coefficient $\frac{\sigma^2}{2}S^2$ is state dependent. Its value may become very small when S is close to zero. Small value of diffusion coefficient may force the use of small step width in explicit schemes due to numerical stability consideration.

- One may prefer uniform step width in the actual asset price S, like increment ΔS of \$1 rather than uniform step width in $\ln S$. The increment Δx corresponds to the proportional jump $e^{\Delta x}$ in the asset price since $\Delta x + \ln S = \ln e^{\Delta x} + \ln S = \ln Se^{\Delta x}$. Note that proportional jumps on the asset price are adopted in the binomial/trinomial tree calculations.

Discretization of the domain

Transform the domain of definition of the continuous problem

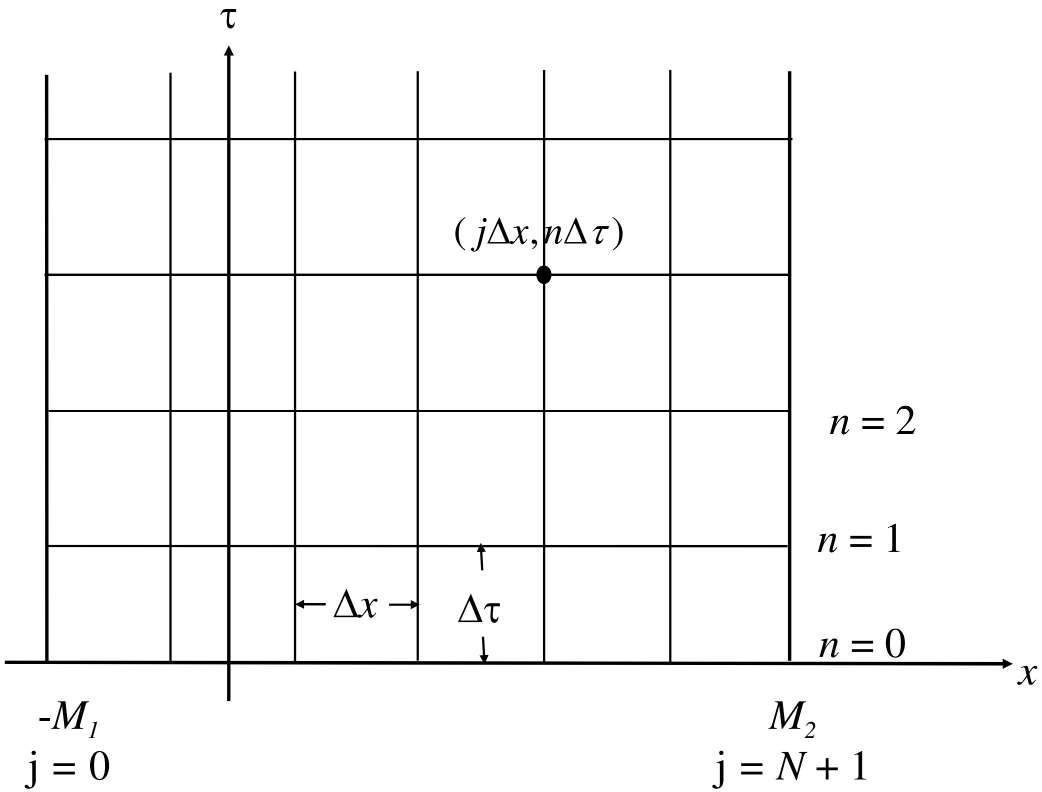
$$\{(x, \tau) : -\infty < x < \infty, \quad \tau \geq 0\}$$

into a discretized domain.

Infinite domain of $x = \ln S$ is approximated by a finite truncated interval $[-M_1, M_2], M_1$ and M_2 are sufficiently large. The discretized domain is overlaid with a uniform system of meshes $(j\Delta x, n\Delta\tau), j = 0, 1, \cdots, N+1, n = 0, 1, 2, \cdots$ with $(N+1)\Delta x = M_1 + M_2$.

Step width Δx and time step $\Delta\tau$ are in general independent. Option values are computed only at the grid points. To reflect the Brownian nature of the asset price process, it is common to choose $\Delta\tau = O(\Delta x^2)$.

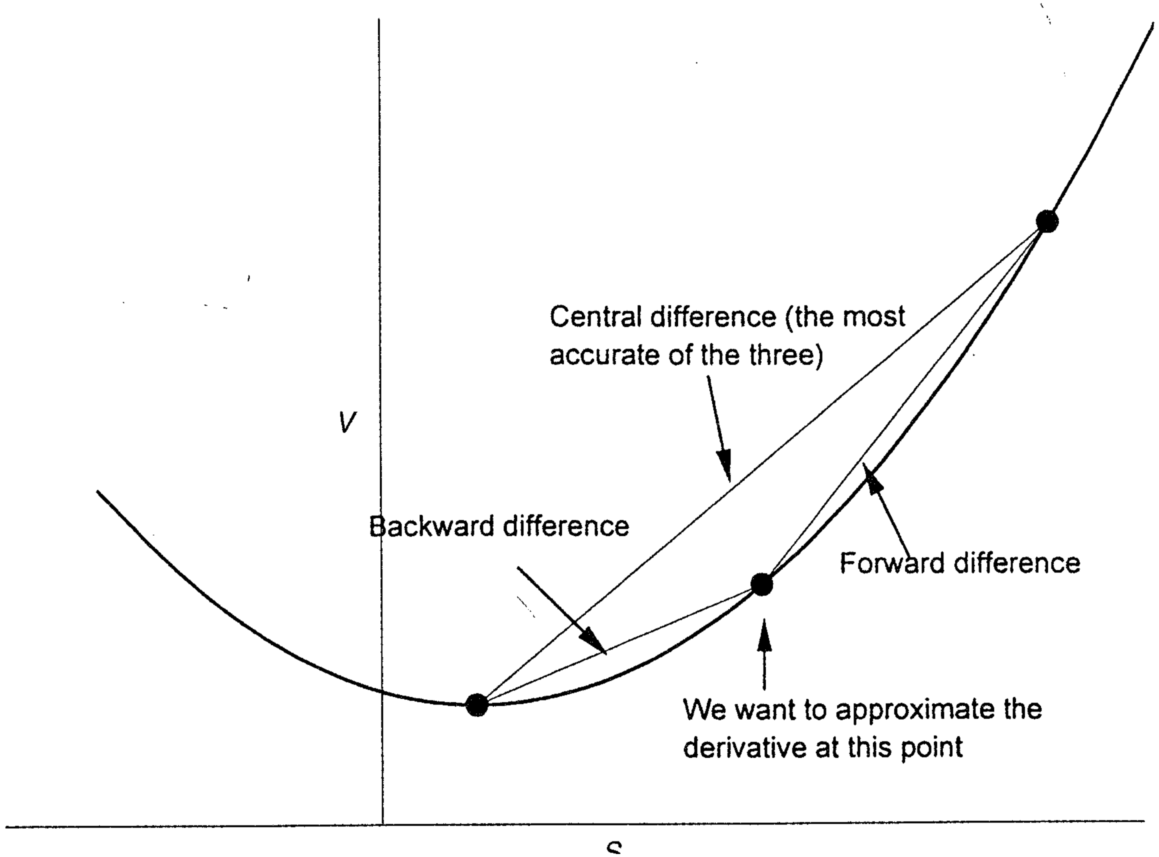
While we perform backward induction in trinomial calculations (going backwards in calendar time), we march forward in the temporal variable τ (time to expiry) in the finite difference calculations.



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.

Respective forward difference, backward difference and centered difference formula at the $(j\Delta S, n\Delta\tau)$ node:

$$\frac{V_{j+1}^n - V_j^n}{\Delta S}, \frac{V_j^n - V_{j-1}^n}{\Delta S} \quad \text{and} \quad \frac{V_{j+1}^n - V_{j-1}^n}{2\Delta S}.$$



Approximations to the delta or $\dfrac{\partial V}{\partial S}$.

Why does centered difference achieve higher order of accuracy compared to forward difference or backward difference?

Consider the centered difference approximation

$$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x},$$

by performing the Taylor expansion of $f(x+\Delta x)$ and $f(x-\Delta x)$, we obtain

$$\begin{aligned}
&\frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \\
= \; & \left\{ \left[f(x) + f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 + \frac{f'''(x)}{3!}\Delta x^3 + \frac{f''''(x)}{4!}\Delta x^4 + \cdots \right] \right. \\
& \left. - \left[f(x) - f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 - \frac{f'''(x)}{3!}\Delta x^3 + \frac{f''''(x)}{4!}\Delta x^4 + \cdots \right] \right\} \Big/ (2\Delta x) \\
= \; & f'(x) + \frac{f'''(x)}{6}\Delta x^2 + \cdots
\end{aligned}$$

so that

$$\frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} = f'(x) + \frac{f'''(x)}{6}\Delta x^2 + O(\Delta x^4).$$

For the forward difference approximation:

$$\frac{f(x+\Delta x)-f(x)}{\Delta x}=f'(x)+\frac{f''(x)}{2}\Delta x+O(\Delta x)^2$$

so that it approximates $f'(x)$ only up to $O(\Delta x)$ accuracy.

In order to achieve $O(\Delta x^2)$ using forward difference, it is necessary to include 3 points, where

$$f'(x)\approx\frac{-f(x+2\Delta x)+4f(x+\Delta x)-3f(x)}{2\Delta x}+O(\Delta x^2).$$

The corresponding 3-point backward difference formula can be deduced to be

$$f'(x)\approx\frac{f(x-2\Delta x)-4f(x-\Delta x)+3f(x)}{2\Delta x}+O(\Delta x^2).$$

Centered difference formula for the second order derivative

$$
\begin{aligned}
f''(x) &\approx \frac{f'\left(x+\frac{\Delta x}{2}\right) - f'\left(x-\frac{\Delta x}{2}\right)}{\Delta x} \\
&\approx \frac{[f(x+\Delta x) - f(x)] - [f(x) - f(x-\Delta x)]}{\Delta x^2} \\
&= \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2}.
\end{aligned}
$$

With symmetry in the centered difference scheme, we are able to achieve $O(\Delta x^2)$ accuracy using only 3 points.

Forward difference formula for the second order derivative

To achieve second order accuracy, we need to use 4 points:

$$f''(x) = \alpha_0 f(x) + \alpha_1 f(x+\Delta x) + \alpha_2 f(x+2\Delta x) + \alpha_3 f(x+3\Delta x) + O(\Delta x^2).$$

We expand $f(x + j\Delta x), j = 1, 2, 3$, at x, and equate the coefficient of $f(x), f'(x)$ and $f'''(x)$ to be zero and the coefficient of $f''(x)$ to be one. The leading error term would be $O(\Delta x^2)$ and involving $f''''(x)$.

$$\begin{aligned}
&\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 = 0 \\
&\alpha_1 + 2\alpha_2 + 3\alpha_3 = 0 \\
&(\alpha_1 + 4\alpha_2 + 9\alpha_3)(\Delta x^2/2) = 1 \\
&\alpha_1 + 8\alpha_2 + 27\alpha_3 = 0.
\end{aligned}$$

We obtain the forward difference formula:

$$f''(x) \approx \frac{2f(x) - 5f(x+\Delta x) + 4f(x+2\Delta x) - f(x+3\Delta x)}{\Delta x^2} + O(\Delta x^2).$$

Similarly, the backward difference formula is given by

$$f''(x) \approx \frac{2f(x) - 5f(x-\Delta x) + 4f(x-2\Delta x) - f(x-3\Delta x)}{\Delta x^2} + O(\Delta x^2).$$

Explicit schemes

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

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

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

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

In terms of W_j^n, by substituting the corresponding difference approximations into the differential equation for W, we have

$$\frac{W_j^{n+1} - W_j^n}{\Delta\tau} = \frac{\sigma^2}{2}\frac{W_{j+1}^{n} - 2W_j^n + W_{j-1}^n}{\Delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{W_{j+1}^{n} - W_{j-1}^n}{2\Delta x}.$$

By observing

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

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

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

- Suppose we are given the "initial" values $V_j^0, j = 0, 1, \cdots, N+1$, along the zeroth time level, we can use the explicit scheme to find values $V_j^1, j = 1, 2, \cdots, N$ along the first time level at $\tau = \triangle\tau$. Forward difference instead of centered difference is used in approximating $\dfrac{\partial V}{\partial \tau}$ since we prefer two-level scheme to three-level scheme.

- 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.

Two-level four-point explicit schemes

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

The above FTCS scheme corresponds to

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

This resembles the trinomial scheme by observing

$$\frac{1}{\lambda^2} = \frac{\sigma^2 \Delta\tau}{\Delta x^2}.$$

For example, b_1 becomes $\frac{1}{2\lambda^2} + \frac{(r - \frac{\sigma^2}{2})\sqrt{\Delta\tau}}{2\lambda\sigma}$, which equals the probability of up-jump p_1 in the trinomial scheme.

In order to avoid the occurrence of negative values in the coefficients, we must observe

(i) $\sigma^2 \frac{\Delta\tau}{\Delta x^2} < 1$
Accordingly, the time step $\Delta\tau$ must be chosen such that $\Delta\tau < \Delta x^2/\sigma^2$.

(ii) $\frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta x^2} > \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{2\Delta x} \Leftrightarrow \Delta x < \frac{\sigma^2/2}{r - \sigma^2/2}$
When $\frac{\sigma^2}{2}$ is small or $r - \frac{\sigma^2}{2}$ is significant (convection dominated), this condition places a stringent constraint on Δx.

The coefficients b_{-1}, b_0 and b_1 have the interpretation of probabilities in the trinomial schemes. Numerical oscillations with unbounded growth will be resulted in the finite difference calculations when some of the probability values are negative.

Both the binomial and trinomial schemes are members of the above family when the tree symmetry condition $ud = 1$ holds.

The up-jump in $x = \ln S$ is given by $\ln u$ in the binomial scheme while the corresponding up-jump in x in the finite difference scheme is Δx, so that $\triangle x = \ln u$. Similarly, $\ln d = -\triangle x$. Note that $x + \Delta x = \ln S + \ln u = \ln uS$ and $x - \Delta x = \ln dS$. The binomial scheme can be expressed as

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

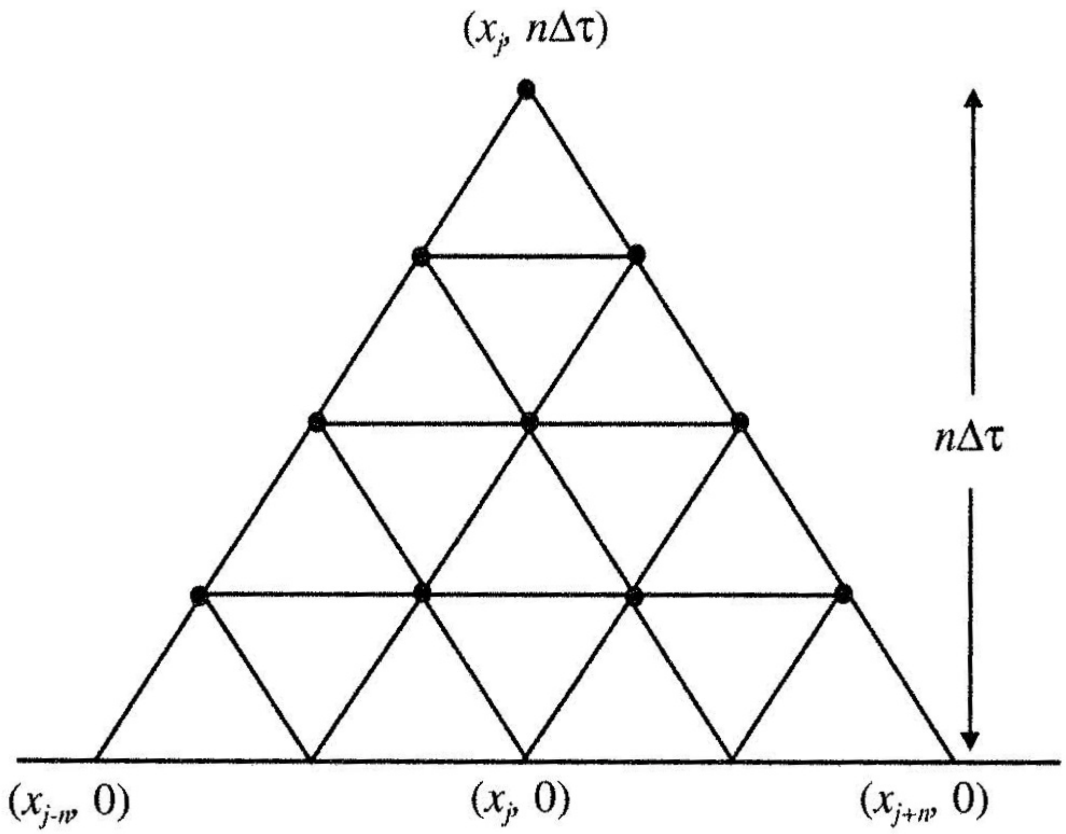
where $V^{n+1}(x), V^n(x + \triangle x)$ and $V^n(x - \triangle x)$ are analogous to $c, c_u^{\Delta t}$ and $c_d^{\Delta t}$, respectively. This corresponds to

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

In the Cox-Ross-Rubinstein scheme, we have $\Delta x = \ln u = \sigma\sqrt{\Delta\tau}$ or $\sigma^2\Delta\tau = \Delta x^2$. 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, provided $\lambda \geq 1$.

Domain of dependence

The lattice tree calculations confine computation of option values within a triangular domain of dependence. This may be seen to be more efficient when single option value at given values of S and τ is required.



The domain of dependence of a binomial scheme with n time steps to expiry.

- The width of the domain of dependence $= 2n\Delta x = 2n \ln u = 2n\sigma\sqrt{\Delta\tau} = 2\sigma\sqrt{T}\sqrt{n}$, where $T = n\Delta\tau$.

- With respect to $x = \ln S$, the width of the domain of dependence of a binomial scheme can be shown to be $O(\sqrt{n})$, where n is the total number of time steps. That is, the width is doubled when the number of time steps is increased by 4-fold. Theoretically, the domain of dependence covers the whole infinite domain $(-\infty, \infty)$ when $n \to \infty$.

- The width of the domain of definition of the continuous European vanilla option model is infinite while that of a barrier option is semi-infinite (one-sided barrier) or finite (two-sided barriers).

Incorporation of boundary conditions by finite difference schemes

- The values at the boundary nodes are dictated by the boundary conditions of the option model. 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 negligence of the boundary conditions does not significantly affect 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 the barrier option models. To achieve sufficient numerical accuracy, it is important that the numerical scheme takes into account the effect of boundary conditions.

- An up-and-out put option with an upstream knock-out barrier B that is continuously monitored would have its domain of definition defined for $-\infty < x < \ln B$. In general, a rebate is paid upon knock-out so that the barrier put option value equals the rebate value upon knock-out. That is,

$$p_{barrier}(\ln B, \tau) = R(\tau),$$

where $R(\tau)$ is the time dependent rebate function.

We must specify the option value along the two boundaries of the computational domain. The boundary conditions specified would depend on the type of option we are solving.

Here, we use S as the independent state variable.

1. To price a call option; at $S = 0, V_0^n = 0$.

   For large S, the call value tends to $S - Xe^{-r(T-t)}$.

   $$V_{N+1}^n = (N+1)\Delta S - Xe^{-rn\Delta\tau}.$$

2. For a put option, at $S = 0, V = Xe^{-r(T-t)}$ so that

   $$V_0^n = Xe^{-rn\Delta\tau}.$$

   The put option becomes worthless for large S so that

   $$V_{N+1}^n = 0.$$

3. When the option has a payoff that is almost linear in the underlying for large values of S, then you can use the upper boundary condition

$$\frac{\partial^2 V}{\partial S^2}(S,t) \to 0 \quad \text{as } S \to \infty.$$

Almost all common derivative contracts have this property. This is particularly useful because it is independent of the type of a contract being valued. We set $\dfrac{\partial^2 V}{\partial S^2}$ along the nodes at $j = N+1$ to be zero. Using the backward difference formula:

$$\left.\frac{\partial^2 V}{\partial S^2}\right|_{(N+1,n)} \approx \frac{2V_{N+1}^n - 5V_N^n + 4V_{N-1}^n - V_{N-2}^n}{\Delta S^2} = 0$$

so that

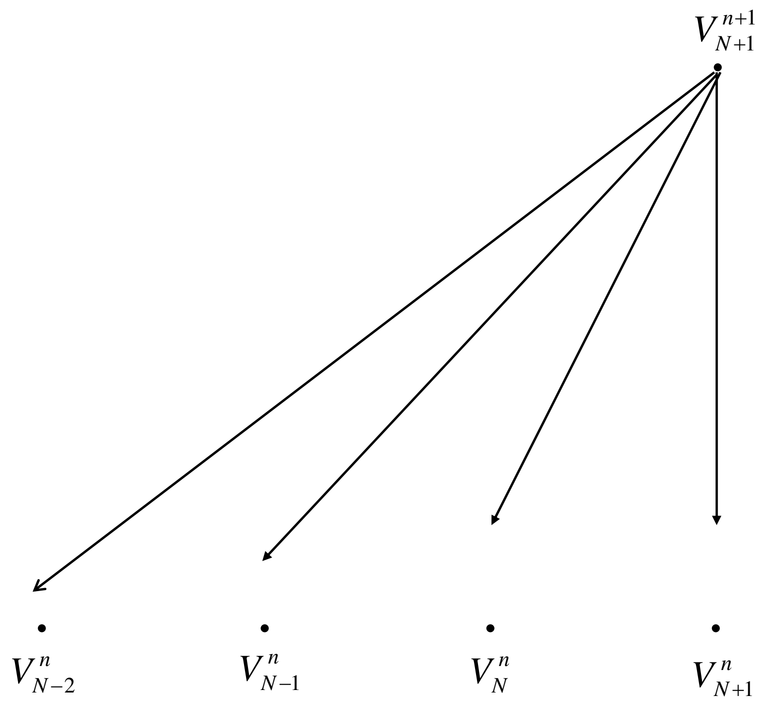
$$V_{N+1}^n = \frac{5V_N^n - 4V_{N-1}^n + V_{N-2}^n}{2}.$$

We obtain the boundary value V_{N+1}^n in terms of the interior values V_N^n, V_{N-1}^n and V_{N-2}^n.

Skew computational scheme with one-sided difference formulas

Computational domain $= \{(x_j, \tau_n) : j = 0, 1, \cdots, N+1, n = 0, 1, 2, \cdots\}$.

Domain of definition of the continuous European vanilla option model $= \{(x, \tau) : -\infty < x < \infty, 0 \leq \tau \leq T\}$



$j = N + 1$ corresponds to the boundary nodes along the right boundary of the computational domain.

Approximation of the Black-Scholes equation at boundary nodes

The option values at $j = N + 1$ should not be prescribed by any boundary conditions arising from the continuous European vanilla option model. If otherwise, it would create undesirable numerical errors if we arbitrarily set inappropriate numerical boundary values.

Rather, we take the notion that the option values at $j = N+1$ should remain to be governed by the Black-Scholes equation.

Recall that we cannot approximate $\frac{\partial^2 V}{\partial x^2}|_{j=N+1}$ using V_N, V_{N+1} and V_{N+2} based on the centered difference formula since node V_{N+2} *does not* exist (outside the computational domain).

We discretize the Black-Scholes equation using the one-sided backward difference formula at the $(N+1)^{\text{th}}$ node along the right boundary:

$$\left.\frac{\partial V}{\partial x}\right|_{j=N+1} \approx \frac{V_{N-1} - 4V_N + 3V_{N+1}}{2\Delta x};$$

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_{j=N+1} \approx \frac{V_{N+1} - 5V_N + 4V_{N-1} - V_{N-2}}{\Delta x^2}$$

so that

$$\begin{aligned}\frac{V_{N+1}^{n+1} - V_{N+1}^n}{\Delta \tau} &= \left(r - \frac{\sigma^2}{2}\right)\frac{V_{N-1}^n - 4V_N^n + 3V_{N+1}^n}{2\Delta x} \\ &\quad + \frac{\sigma^2}{2}\frac{2V_{N+1}^n - 5V_N^n + 4V_{N-1}^n - V_{N-2}^n}{\Delta x^2} - rV_{N+1}^n.\end{aligned}$$

V_{N+1}^{n+1} can be determined from known values of $V_{N-2}^n, V_{N-1}^n, V_N^n$ and V_{N+1}^n at the n^{th} time level.

Advantages of explicit schemes

- It is easier to program and less likely to make mistakes.
- When it does go unstable it is usually obvious.
- It is easy to incorporate accurate one-sided differences.

Disadvantages of explicit schemes

- There are restrictions on the time step due to numerical stability consideration so the method would be less efficient than the implicit schemes (to be discussed later).
- The information of the boundary conditions takes a finite number of time steps to take effect into option valuation at nodes that are far from the boundary.

Crank-Nicolson scheme

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

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

and the temporal derivative by the centered difference

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

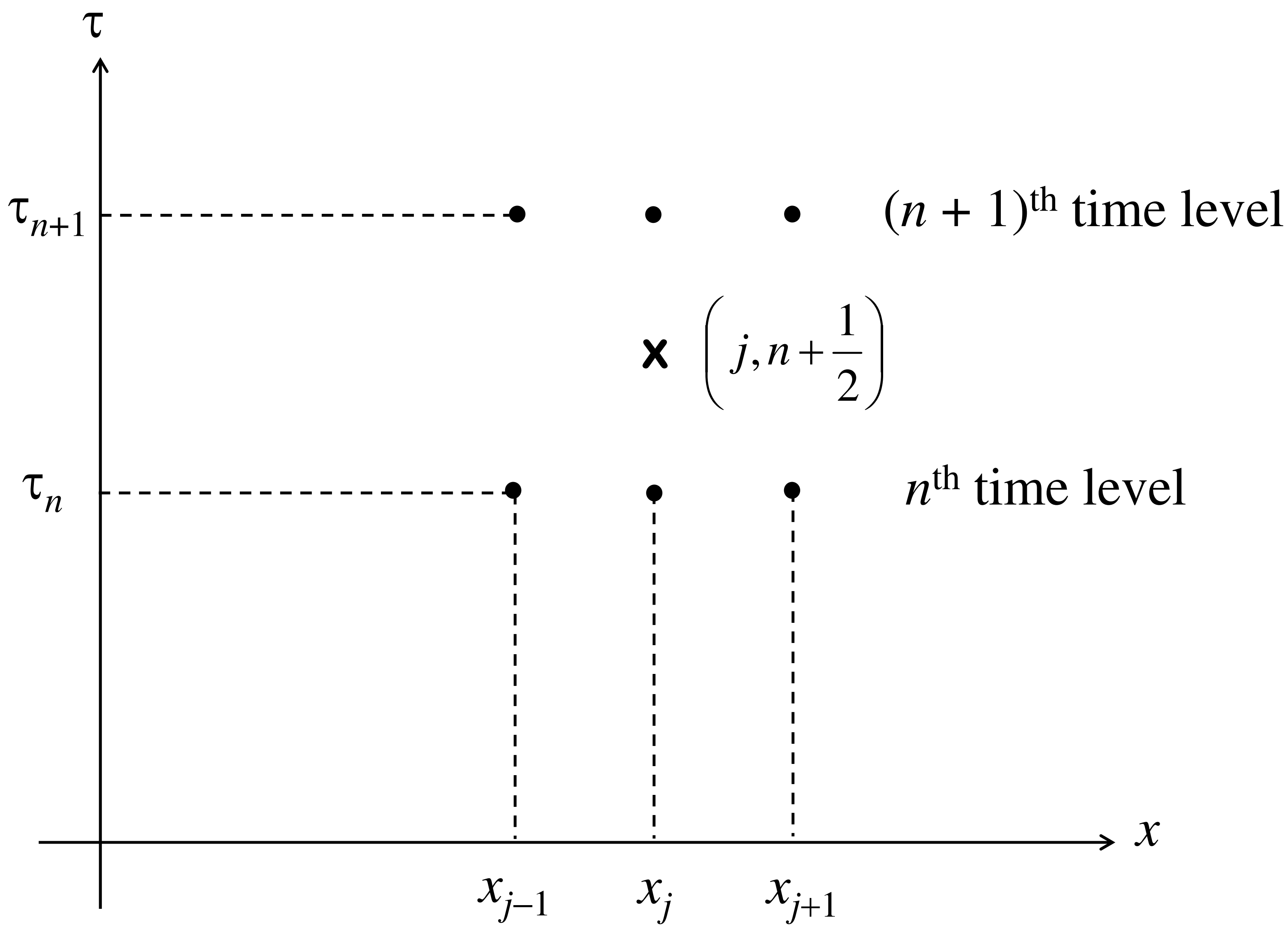
we then obtain the following two-level implicit finite difference scheme

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

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 that take the form

$$
a_1 V_{j+1}^{n+1} + a_0 V_j^{n+1} + a_{-1} V_{j-1}^{n+1} = b_1 V_{j+1}^n + b_0 V_j^n + b_{-1} V_{j-1}^n, \qquad j = 1, 2, \cdots, N, \quad n = 0, 1, \cdots .
$$



The Crank-Nicolson scheme involves 3 option values at both the n^{th} and $(n+1)^{\text{th}}$ time level.

In order to achieve $O(\Delta\tau^2)$ accuracy, we approximate $V, \dfrac{\partial V}{\partial \tau}, \dfrac{\partial V}{\partial x}, \dfrac{\partial^2 V}{\partial x^2}$ at the fictitious intermediate $\left(n + \dfrac{1}{2}\right)^{\text{th}}$ time level.

$$\left.\frac{\partial^2 V}{\partial x^2}\right)_{j,n+\frac{1}{2}} \approx \frac{1}{2}\left[\left.\frac{\partial^2 V}{\partial x^2}\right|_{j,n+1} + \left.\frac{\partial^2 V}{\partial x^2}\right|_{j,n}\right]$$

$$\left.\frac{\partial V}{\partial \tau}\right)_{j,n+\frac{1}{2}} \approx \frac{V_j^{n+\frac{1}{2}+\frac{1}{2}} - V_j^{n+\frac{1}{2}-\frac{1}{2}}}{2\left(\frac{\Delta\tau}{2}\right)} = \frac{V_j^{n+1} - V_j^n}{\Delta\tau}.$$

Relate V_{j+1}^{n+1}, V_j^{n+1} and V_{j-1}^{n+1} (to be computed at the new time level) with V_{j+1}^n, V_j^n and V_{j-1}^n (known values at the old time level).

Suppose the 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. For the simple cases, we may assume V_0^{n+1} and V_{N+1}^{n+1} to be known values directly available from the boundary conditions. The two-level six-point scheme can be represented as

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

where

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

Thomas algorithm – solution of the tridiagonal system

Consider the solution of the following tridiagonal system of the form

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

with $V_0 = V_{N+1} = 0$. For the more general consideration, we allow the coefficients to differ among equations. The imposition of $V_0 = V_{N+1} = 0$ dictates that the first and the last equations have only 2 unknowns.

- In the first step of elimination, we reduce the system to the upper triangular form by eliminating V_{j-1} in the j^{th} equation, $j = 2, 3, \ldots, N$.

- 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.

- We express V_2 in terms of V_3 and some known quantities. We then substitute into the third equation, ..., and so on.

- At the end of the elimination procedure, the last but one equation and the last equation both have only 2 unknowns. They can be solved easily to obtain V_{N-1} and V_N.

- Once V_{N-1} is available, since the last but two equation has been reduced to contain V_{N-2}, V_{N-1} only, the solution to V_{N-2} can then be obtained directly. We proceed to obtain $V_{N-3}, V_{N-4}, \cdots, V_2, V_1$ by successive backward substitution.

Suppose the first k equations have been reduced to the form

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

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

Gaussian elimination procedure at work

We use the reduced form of the k^{th} equation

$$V_k - e_k V_{k+1} = f_k$$

and the original $(k+1)^{\text{th}}$ equation

$$-a_{k+1}V_k + b_{k+1}V_{k+1} - c_{k+1}V_{k+2} = d_{k+1}$$

to obtain the new $(k+1)^{\text{th}}$ reduced equation

$$V_{k+1} - e_{k+1}V_{k+2} = f_{k+1}.$$

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

$$\begin{aligned} & -a_{k+1}(e_k V_{k+1} + f_k) + b_{k+1}V_{k+1} - c_{k+1}V_{k+2} = d_{k+1} \\ \Leftrightarrow \quad & V_{k+1} - \frac{c_{k+1}}{b_{k+1} - a_{k+1}e_k} V_{k+2} = \frac{d_{k+1} + a_{k+1}f_k}{b_{k+1} - a_{k+1}e_k}. \end{aligned}$$

We then deduce the following recurrence relations for e_j and f_j:

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

The first equation is

$$b_1 V_1 - c_1 V_2 = d_1 \quad \text{or} \quad V_1 - \frac{c_1}{b_1} V_2 = \frac{d_1}{b_1}$$

so that $e_1 = c_1/b_1$ and $f_1 = d_1/b_1$. Apparently, we may start with $e_0 = f_0 = 0$ in the recurrence relation and obtain the same set of values for e_1 and f_1. The tridiagonal system is now effectively reduced to a bidiagonal (upper diagonal) form.

Starting from the above initial values, the recurrence relations can be used to find all values e_j and $f_j, j = 1, 2, \cdots, N$. 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 $V_{N+1} = 0$. That is, $V_N = f_N$, and $V_{N-1} = e_{N-1} V_N + f_{N-1}$, etc.

Summary

1. Compute e_j and f_j, $j = 1, 2, \cdots, N$, using the following recursive relations:

$$
\begin{aligned}
e_j &= \tfrac{c_j}{b_j - a_j e_{j-1}}, \quad e_0 = 0; \\
f_j &= \tfrac{d_j + a_j f_{j-1}}{b_j - a_j e_{j-1}}, \quad f_0 = 0.
\end{aligned}
$$

2. Solve backward for V_N, $V_{N-1}, \cdots, V_1$, where

$$V_k - e_k V_{k+1} = f_k, \quad k = N-1, N-2, \cdots, 1 \quad \text{with} \quad V_N = f_N.$$

- On the control of the growth of roundoff errors, 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. This condition would pose certain constraint on the choice of $\Delta\tau$ and Δx in the Crank-Nicolson scheme.

Computational efficiency

- The Thomas algorithm is a very efficient algorithm where the tridiagonal system can be solved with 4 (add/subtract) and 6 (multiply/divide) operations per node point.

- More precisely, we need 2 multiply/divide and 1 add/subtract in calculating e_j, 3 multiple/divide and 2 add/subtract in calculating f_j, 1 multiply/divide and 1 add/subtract in calculating V_j.

- Compared with the explicit schemes (which requires 3 multiply/divide and 2 add/subtract), it takes about twice the number of operations per time step.

Incorporation of boundary conditions

If the option values are not prescribed on the boundaries of the computational domain, then the first and the last equation may contain 4 unknowns due to the choice of skew computational stencil at the boundary nodes. Consider the left boundary node at $j = 0$, we have the equation that involves V_0, V_1, V_2 and V_3, where

$$a_0 V_0 + b_0 V_1 + c_0 V_2 + d_0 V_3 = f_0.$$

For the next node, $j = 1$, 3 unknowns are involved where

$$a_1 V_0 + b_1 V_1 + c_1 V_2 = f_1.$$

For $j = 2$, we have

$$a_2 V_1 + b_2 V_2 + c_2 V_3 = f_2.$$

From the above 3 equations, we can eliminate V_0 and V_1 to obtain an equation that involves V_2 and V_3.

One can then proceed with the Thomas algorithm as usual.

Fully implicit scheme

Suppose we approximate $V(j\Delta x, (n+1)\Delta\tau)$ using backward difference in $\dfrac{\partial V}{\partial \tau}$ and centered difference in $\dfrac{\partial V}{\partial x}$ and $\dfrac{\partial^2 V}{\partial x^2}$, we obtain the following implicit scheme:

$$\frac{V_j^{n+1} - V_j^n}{\Delta\tau} = \frac{\sigma^2}{2}\frac{V_{j+1}^{n+1} - 2V_j^{n+1} + V_{j-1}^{n+1}}{\Delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2\Delta x} - rV_j^{n+1}.$$

This leads to the following two-level-four-point scheme

$$a_1 V_{j+1}^{n+1} + a_0 V_j^{n+1} + a_{-1} V_{j-1}^{n+1} = V_j^n$$

where

$$\begin{aligned}
a_1 &= -\left[\frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{2\Delta x}\right] \\
a_0 &= 1 + \sigma^2\frac{\Delta\tau}{\Delta x^2} + r \\
a_{-1} &= -\left[\frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta x^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{2\Delta x}\right].
\end{aligned}$$

Advantages

1. Retain all the advantages of implicit schemes, like numerical stability, immediate response to boundary conditions (through solution of a tridiagonal system of equations at every time step).

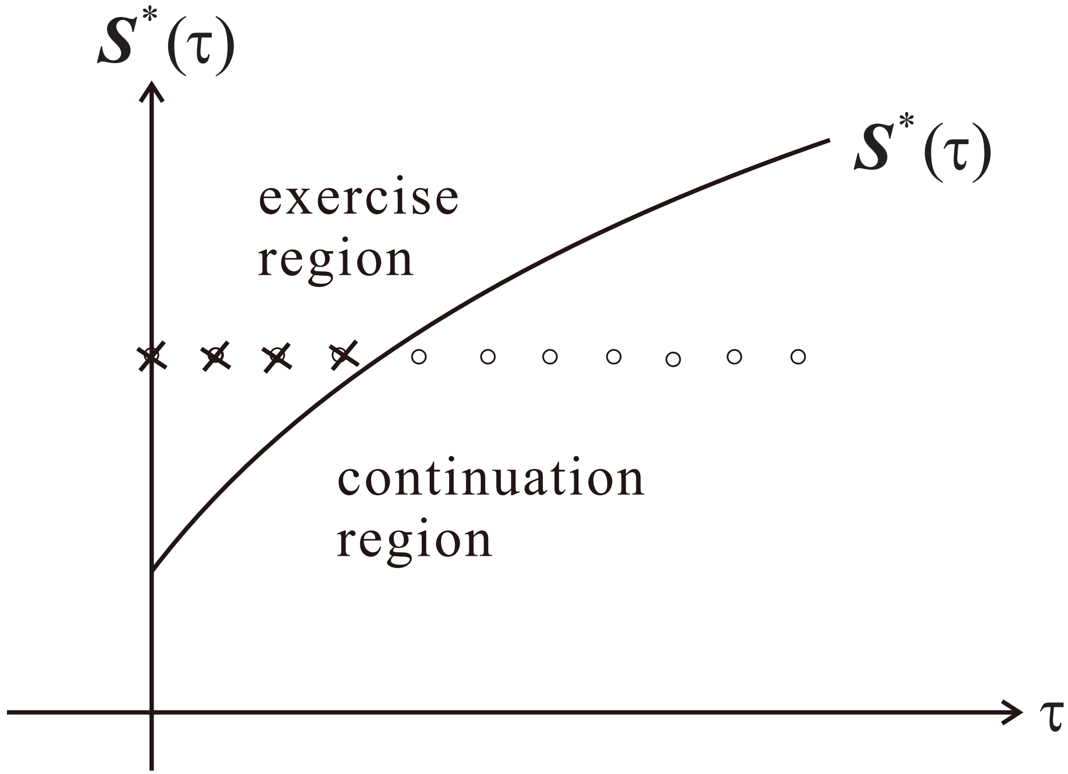
2. Avoidance of potential numerical oscillations commonly called the Crank-Nicolson noises.

3. The adoption of backward difference in $\dfrac{\partial V}{\partial \tau}$ would lead to $O(\Delta\tau)$ accuracy. However, since we typically keep $\Delta\tau = O(\Delta x^2)$, so the overall $O(\Delta x^2)$ accuracy is maintained. Recall that we take $\sigma^2\Delta\tau = \lambda^2\Delta x^2$ in the trinomial scheme and $\Delta x = \ln u = \sigma\sqrt{\Delta\tau}$ in the binomial scheme.

Iterated method for pricing American options

The naive approach of computing V_j^{n+1} from the tridiagonal system of equations derived from an implicit scheme, then followed by comparing V_j^{n+1} with the intrinsic value is NOT acceptable since we do not know in advance whether the neighboring nodal values V_{j-1}^{n+1} and V_{j+1}^{n+1} assume the intrinsic value or the corresponding continuation value.

In other words, the original tridiagonal system of equations for $\boldsymbol{V}^{n+1} = (V_1^{n+1} \cdots V_N^{n+1})$ is not the appropriate system of equations for the computation of the continuation values. This is because the system of equations have not incorporated the information on whether the option values at neighboring nodes assume the continuation value or exercise value.



Plot of the exercise region and continuation region of an America call option

At a given time level $\tau = n\Delta\tau$, some nodes lie in the continuation region while others lie in the exercise region, so we should not use the same tridiagonal system for solving $\boldsymbol{V}^n$ as in the case of pricing European options.

Gauss-Siedel iterative scheme for numerical solution of linear system of equations

Consider an algebraic system of equations

$$A\boldsymbol{x} = \boldsymbol{b},$$

we write $A = D + L + U$, where L is the lower triangular part and U is the upper triangular part of A, respectively.
We start with $(L+D)\boldsymbol{x} = -U\boldsymbol{x}+\boldsymbol{b}$, and construct the iterative scheme:

$$(L + D)\boldsymbol{x}^{(k)} = -U\boldsymbol{x}^{(k-1)} + \boldsymbol{b}$$

or

$$\boldsymbol{x}^{(k)} = D^{-1}(\boldsymbol{b} - L\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k-1)}).$$

For the i^{th} component of $\boldsymbol{x}^{(k)}$, the Gauss-Seidel scheme is

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} A_{ij}x_j^{(k-1)}\right).$$

The computational of $x_i^{(k)}$ uses $x_j^{(k)}$, $j = 1, 2, \ldots, i-1$, that have already been computed. For x_j, $j = i+1, \ldots, n$, only the $(k-1)^{\text{th}}$ iterates are known.

Conversion of an implicit evaluation into explicit calculations via an iterative procedure

Consider an implicit finite difference scheme of the form

$$a_{-1}V_{j-1} + a_0 V_j + a_1 V_{j+1} = d_j, \quad j = 1, 2, \cdots, N,$$

where the superscript "$n+1$" is omitted for brevity, and d_j represents the known quantities. We express V_j in terms of other quantities as follows:

$$V_j = \frac{1}{a_0}\left(d_j - a_{-1}V_{j-1} - a_1 V_{j+1}\right).$$

The *Gauss-Seidel* iteration produces the k^{th} iterate of V_j by

$$\begin{aligned} V_j^{(k)} &= \frac{1}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_1 V_{j+1}^{(k-1)}\right) \\ &= V_j^{(k-1)} + \frac{1}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_0 V_j^{(k-1)} - a_1 V_{j+1}^{(k-1)}\right), \end{aligned}$$

where the last term in the above equation represents the correction made on the $(k-1)^{\text{th}}$ iterate of V_j.

We start from $j = 1$ and proceed sequentially with increasing value of j. When we compute $V_j^{(k)}$ in the k^{th} iteration, the new k^{th} iterate $V_{j-1}^{(k)}$ is already available while only the old $(k-1)^{\text{th}}$ iterate $V_{j+1}^{(k-1)}$ is known.

By proceeding sequentially node by node, the dynamic programming procedure of maximizing the option value by the choice of either continuation or early exercise can be updated immediately based on the most recent iterates.

Applying the dynamic programming procedure at each iteration step

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 given by

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

When the j^{th} node is in the exercise region, V_j takes up h_j. When the j^{th} node is in the continuation region, V_j satisfies the difference scheme at the j^{th} node and will not take up h_j. In summary, the successive iterates of the estimated option values take advantage of the updated information on whether the continuation value or exercise value is adopted by the neighboring nodal option values.

Choices of the starting iterates

For nodal option values at the $(n+1)^{\text{th}}$ time level, based on the crude approximation that $V_j^{n+1} - V_j^n \approx V_j^n - V_j^{n-1}$, it is convenient to choose the zeroth iterate to be $\left(V_j^{n+1}\right)^{(0)} = V_j^n + (V_j^n - V_j^{n-1}) = 2V_j^n - V_j^{n-1}$, $n \geq 1$. For $n = 0$, we may use the Black Scholes pricing formula to approximate the American option values along the first time level.

Termination criterion

A sufficient number of iterations are performed 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 taken to be the numerical approximate solution for V_j.

Discretization of the two-asset Black-Scholes equation

Let $x_1 = \ln S_1$ and $x_2 = \ln S_2$, the option price function $V(x_1, x_2, \tau)$ is governed by

$$\frac{\partial V}{\partial \tau} = \frac{\sigma_1^2}{2}\frac{\partial^2 V}{\partial x_1^2} + \rho\sigma_1\sigma_2\frac{\partial^2 V}{\partial x_1 \partial x_2} + \frac{\sigma_2^2}{2}\frac{\partial^2 V}{\partial x_2^2} + \left(r - \frac{\sigma_1^2}{2}\right)\frac{\partial V}{\partial x_1} + \left(r - \frac{\sigma_2^2}{2}\right)\frac{\partial V}{\partial x_2} - rV,$$

$$-\infty < x_1 < \infty,\ -\infty < x_2 < \infty,\ \tau > 0.$$

How to discretize the second order derivative terms?

$$\left.\frac{\partial^2 V}{\partial x_1^2}\right|_{(i_1,i_2,n)} \approx \frac{V^n_{i_1+1,i_2} - 2V^n_{i_1,i_2} + V^n_{i_1-1,i_2}}{\Delta x_1^2}$$

$(i_1 - 1, i_2)$ • (i_1, i_2) • $(i_1 + 1, i_2)$ •

$$\left.\frac{\partial^2 V}{\partial x_2^2}\right|_{(i_1,i_2,n)} \approx \frac{V^n_{i_1,i_2+1} - 2V^n_{i_1,i_2} + V^n_{i_1,i_2-1}}{\Delta x_2^2}$$

•$(i_1, i_2 + 1)$
•(i_1, i_2)
•$(i_1, i_2 - 1)$

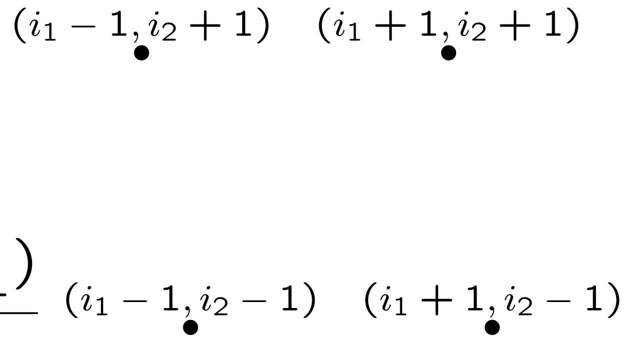
The combination of these two discretized forms involve 5 points in the computational stencil.

We define the centered difference operators ξ_{x_1} and ξ_{x_2} that approximate $\dfrac{\partial}{\partial x_1}$ and $\dfrac{\partial}{\partial x_2}$ as follows:

$$\begin{aligned}
\xi_{x_1} V^n_{i_1,i_2} &= \frac{V^n_{i_1+1,i_2} - V^n_{i_1-1,i_2}}{2\Delta x_1},\\
\xi_{x_2} V^n_{i_1,i_2} &= \frac{V^n_{i_1,i_2+1} - V^n_{i_1,i_2-1}}{2\Delta x_2}.
\end{aligned}$$

We approximate the cross-derivative term at (i_1, i_2, n) by

$$\begin{aligned}
&\left.\frac{\partial^2 V}{\partial x_1 \partial x_2}\right|_{(i_1,i_2,n)}\\
&\approx \xi_{x_2}(\Delta x_1\ V^n_{i_1,i_2})\\
&= \xi_{x_2}\frac{V^n_{i_1+1,i_2} - V^n_{i_1-1,i_2}}{2\Delta x_1}\\
&= \frac{(V^n_{i_1+1,i_2+1} - V^n_{i_1+1,i_2-1}) - (V^n_{i_1-1,i_2+1} - V^n_{i_1-1,i_2-1})}{4\Delta x_1 \Delta x_2}
\end{aligned}$$



The above discretization of the cross-derivative term adds another 4 points in the computational stencil. The Kamrad-Ritchken's trinomial scheme involves only 5 points. Since the finite difference discretization suffers from higher computational complexity when compared with the probabilistic approach in deriving the trinomial scheme, so we adopt the 5-point trinomial scheme instead of the 9-point finite difference scheme.

The 5-point trinomial scheme works well when σ is not state dependent. When σ_i is a function of S_i and t, $i = 1, 2$, the probability values cannot be found easily.

How to derive the skew computational stencil along a boundary of the computational domain?

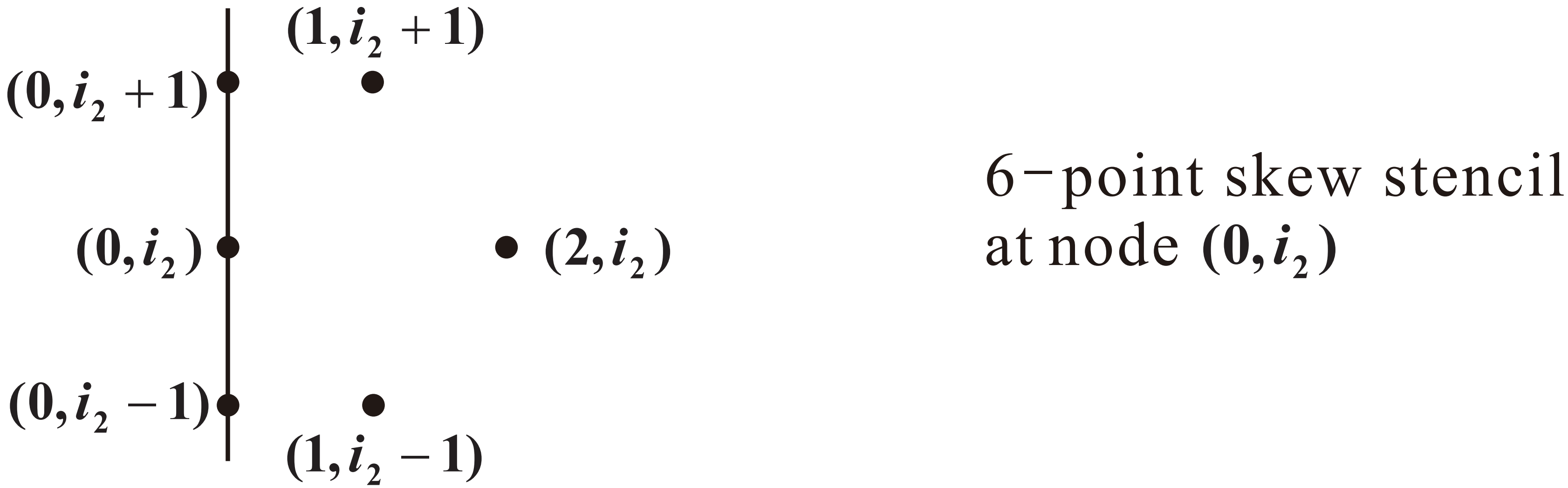
Due to loss of symmetry in a skew stencil, we abandon the 5-point scheme and use the 6-point scheme that involves 6 probability values. We obtain 6 equations for the 6 probability values by

(i) equating two means,

(ii) equating two variances and one covariance,

(iii) sum of probability values equals one.

One natural choice of 6-point stencil involves $(0, i_2+1)$, $(0, i_2)$, $(0, i_2-1)$, $(1, i_2)$, $(2, i_2)$, $(3, i_2)$, that is, taking 4 points along the horizontal row i_2. A more preferable choice is presented below:



We write the discretized scheme in the form

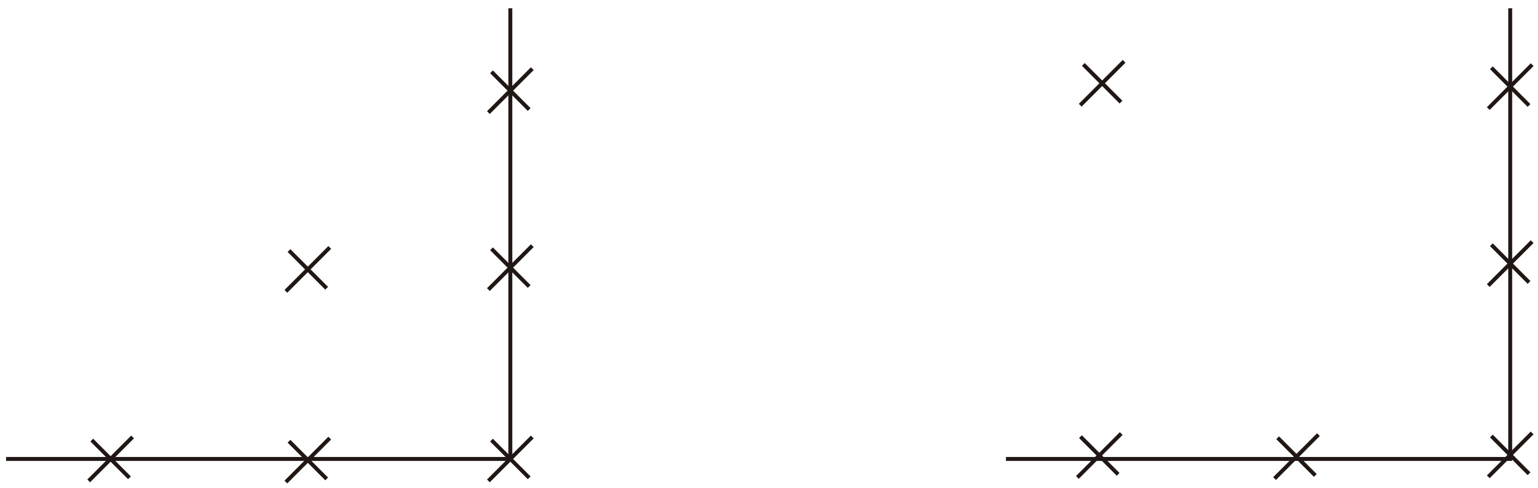
$$V^{n+1}_{0,i_2} = \Big(p_{0,0}V^n_{0,i_2} + p_{0,-1}V^n_{0,i_2-1} + p_{0,1}V^n_{0,i_2+1} + p_{1,-1}V^n_{1,i_2-1} + p_{1,1}V^n_{1,i_2+1} + p_{2,0}V^n_{2,i_2}\Big)\, e^{-r\Delta t}.$$

We use the approximation: var$(X) \approx E(X^2)$ since $E(X)^2$ is $O(\Delta\tau^2)$; the same for cov$(XY) \approx E(XY)$ since $E(X)E(Y)$ is $O(\Delta\tau^2)$. The 6 equations for the probability values are given by

$$\Delta x_1(2p_{2,0} + p_{1,-1} + p_{1,1}) = (r - \frac{\sigma_1^2}{2})\Delta\tau$$
$$\Delta x_2(p_{0,1} - p_{0,-1} + p_{1,1} - p_{1,-1}) = (r - \frac{\sigma_2^2}{2})\Delta\tau$$
$$\Delta x_1^2(4p_{2,0} + p_{1,-1} + p_{1,1}) = \sigma_1^2\Delta\tau$$
$$\Delta x_2^2(p_{0,1} + p_{0,-1} + p_{1,1} + p_{1,-1}) = \sigma_2^2\Delta\tau$$
$$\Delta x_1\Delta x_2(-p_{1,-1} + p_{1,1}) = \rho\sigma_1\sigma_2\Delta\tau$$
$$p_{0,0} + p_{0,-1} + p_{0,1} + p_{1,-1} + p_{1,1} + p_{2,0} = 1.$$

Note that $O(\Delta\tau)^2$ terms are neglected in the calculations of the two variances and one covariance.

How about the choice of stencil of a node at a corner of the computational domain?



Which of the above 6-point scheme would be the better choice?

One criterion is to examine which scheme has less stringent conditions on Δx_1, Δx_2 and Δt with regard to avoidance of negative probability values.

3.2 Numerical approximation of auxiliary conditions

Auxiliary conditions refer to the terminal payoff function plus additional boundary conditions due to the knock-out feature or embedded path dependent features in the option contract.

"Initial" condition is prescribed by the terminal payoff

At expiry, the option value is just the terminal payoff function. We have

$$V(S,T) = \text{Payoff}(S),$$

or in the finite difference setting,

$$V_j^0 = \text{Payoff}(j\Delta S).$$

For example, when pricing a call option, we set

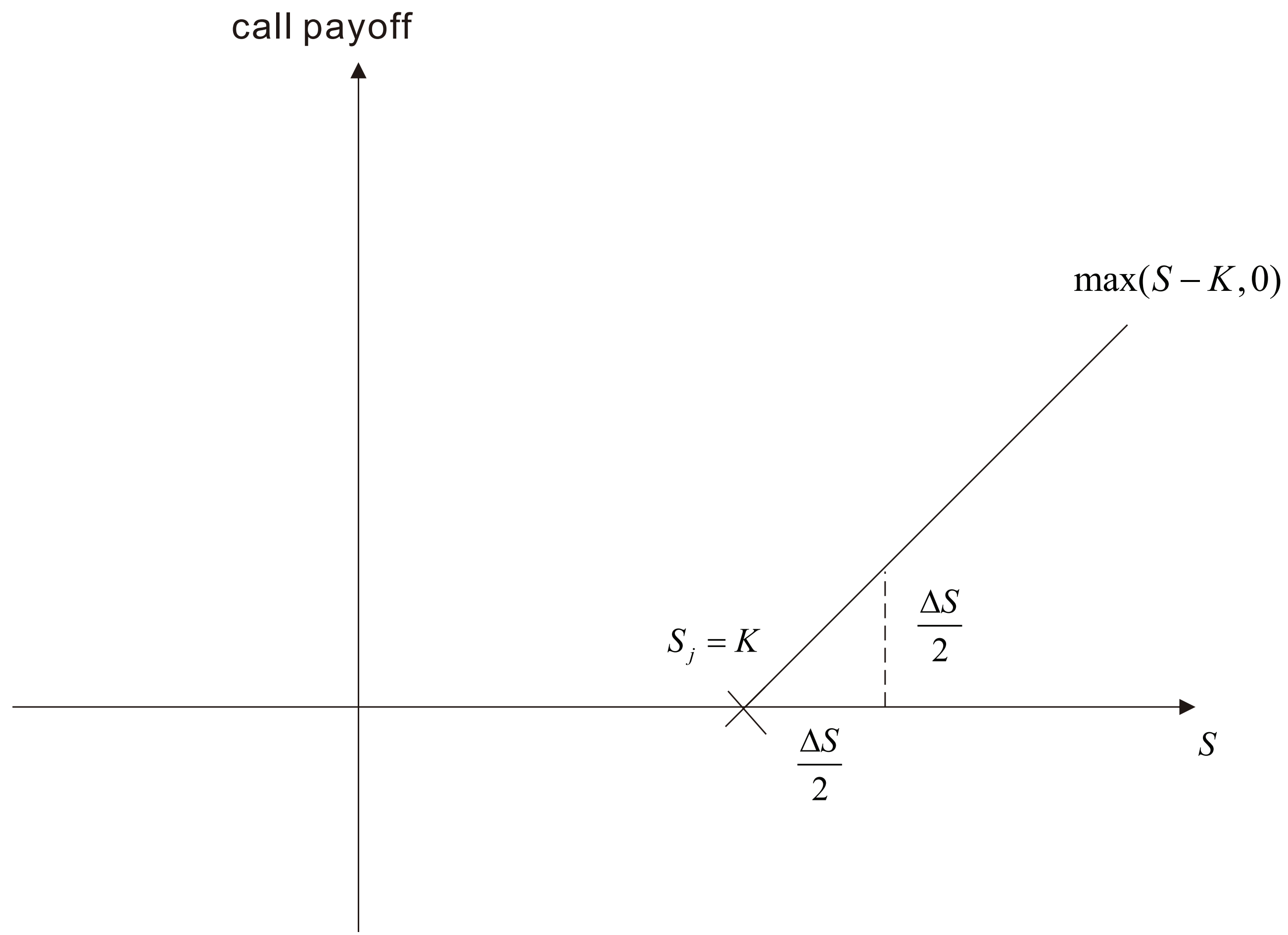
$$V_j^0 = \max\left(j\Delta S - K, 0\right).$$

Smoothing of discontinuities in the terminal payoff functions

- Most terminal payoff functions of options have some form of discontinuity (like the binary payoff) or non-differentiability (like the call or put payoff). Quantization error arises since the payoff function is sampled at discrete node points.

- Let $V_T(S)$ denote the terminal payoff function. Instead of simply taking the value $V_T(S_j)$, the payoff value at node S_j is given by averaging over the node cell, where

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



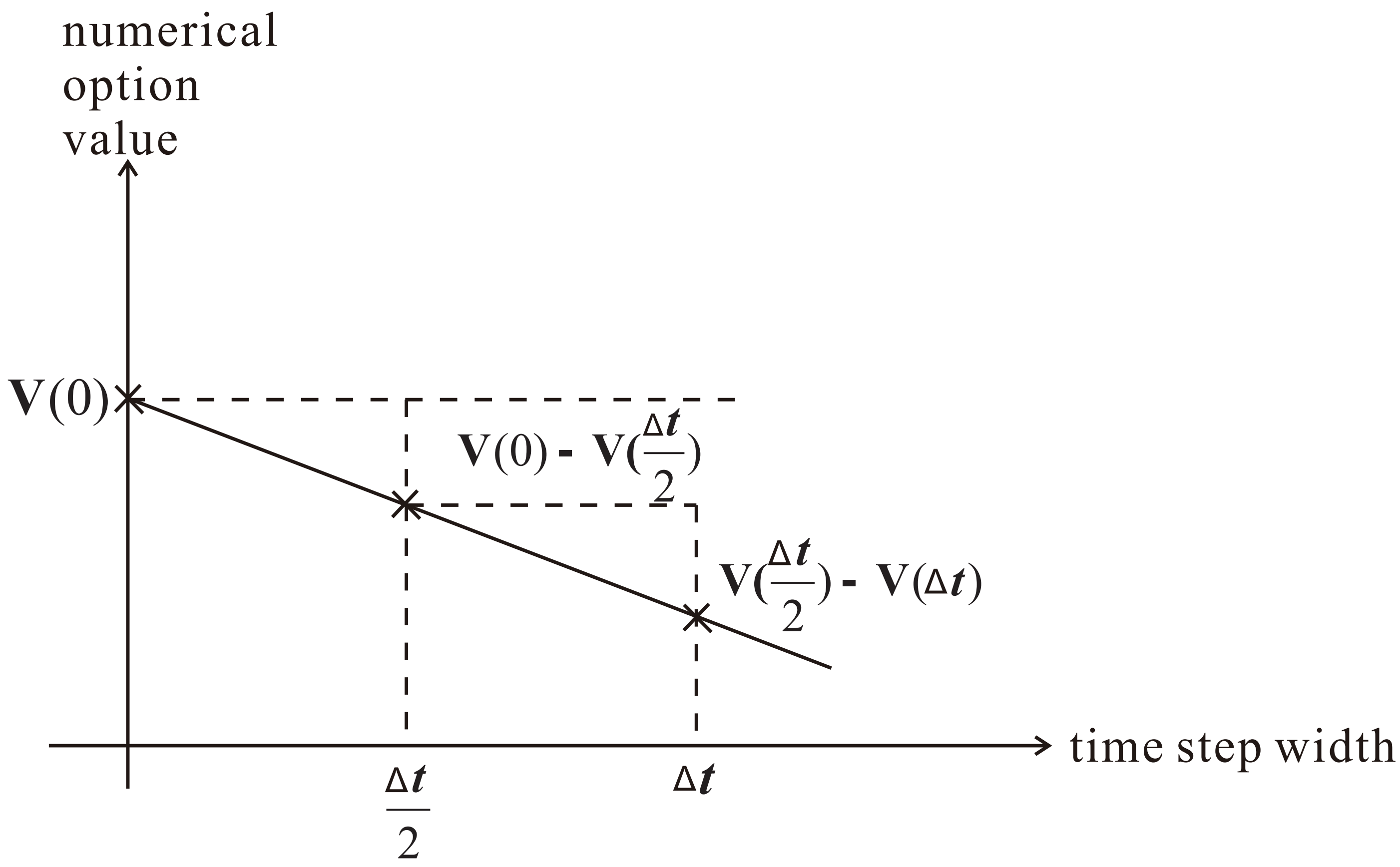
Take the call payoff $\max(S-K,0)$ as an example. Suppose we set the strike price K fall exactly on a node point, then $V_T(S_j)=0$ while the cell-averaged value over $\left[S_j-\frac{\Delta S}{2}, S_j+\frac{\Delta S}{2}\right]$ is $\frac{1}{2}\left(\frac{\Delta S}{2}\right)^2 \Big/ \Delta S = \Delta S/8$.

Averaging the terminal payoff for the vanilla European and American calls can provide a smoother convergence. The smoothed numerical solutions then allow the application of extrapolation for convergence enhancement.

The idea of extrapolation is to find the extrapolated numerical value $V(0)$ at vanishing time step (continuous solution). Suppose the numerical error is reduced by half when the time step is reduced by half (linear rate of convergence), then

$$V(0) \approx V\left(\frac{\Delta t}{2}\right) + \left[V\left(\frac{\Delta t}{2}\right) - V\left(\Delta t\right)\right].$$

The extrapolation procedure works well only if convergence of numerical solution value is not erratic (caused by discontinuity of "initial" condition).



If the linear rate of convergence with respect to Δt is almost observed, we may assume

$$V(0) - V(\frac{\Delta t}{2}) \approx V(\frac{\Delta t}{2}) - V(\Delta t)$$

so that

$$V(0) \approx 2V(\frac{\Delta t}{2}) - V(\Delta t).$$

Black-Scholes approximation

Useful for pricing American options and exotic options for which the Black-Scholes solution is a good approximation at time close to expiry. Use the Black-Scholes values along the first time level and proceed with the usual finite difference calculations at subsequent time levels. Since the Black-Scholes pricing formula involves the cumulative normal distribution functions, smoothing of the first level nodal option values is resulted.

Numerical examples

Consider the valuation of a European call option on a stock whose price is 100. The strike is 100, time to maturity is one year, volatility is 40%, and the continuously compounded annual interest is 6%.

We compare the error ratios when the number of time steps (n) is doubled. Three sets of calculations were performed.

- Binomial calculations
- Finite difference calculations with smoothing of the terminal call payoff
- Finite difference calculations with the Black-Scholes adjustment.

1. Without smoothing of "initial condition" nor Black-Scholes adjustment

n	Exact	Binomial	Error ratio	Extrapolated
10	18.47260446	−0.18552506		
20	18.47260446	−0.05399578	3.43591784	0.07753350
40	18.47260446	−0.00191274	28.22959146	0.05017031
80	18.47260446	0.01410255	−0.13563052	0.03011785
160	18.47260446	0.01496727	0.94222596	0.01583199
320	18.47260446	0.01034461	1.44686708	0.00572194
640	18.47260446	0.00446058	2.31911889	−0.00142345
1280	18.47260446	−0.00100635	−4.43243516	−0.00647328

The columns under "binomial" and "extrapolated" show the numerical errors at the given number of time steps n. Numerical error is defined by numerical approximate value - exact value.

The error ratio at $n = 20$ is calculated by the error at $n = 10$ divided by the error at $n = 20$.

2. Smoothing of the terminal call payoff

n	Exact	Smoothing	Error ratio	Extrapolated
10	18.47260446	0.49641997		
20	18.47260446	0.25124368	1.97585053	0.00606740
40	18.47260446	0.12642076	1.98736102	0.00159783
80	18.47260446	0.06341645	1.99350098	0.00041214
160	18.47260446	0.03176181	1.99662566	0.00010718
320	18.47260446	0.01589584	1.99812100	0.00002987
640	18.47260446	0.00795056	1.99933536	0.00000528
1280	18.47260446	0.00397364	2.00082367	−0.00000327

Let $V(\infty)$ denote the exact solution (with infinite number of time steps) and $V(n)$ denote the numerical approximate value using n time steps. The error ratio $\dfrac{V(n) - V(\infty)}{V(2n) - V(\infty)}$ is close to 2 for most values of n. Under such scenario, the extrapolation procedure works very well to achieve highly accurate numerical approximate solution.

3. Black-Scholes adjustment at the first time level

n	Exact	BS Adjustment	Error ratio	Extrapolated
10	18.47260446	0.08592189		
20	18.47260446	0.04385530	1.95921356	0.00178870
40	18.47260446	0.02210462	1.98398798	0.00035394
80	18.47260446	0.01102834	2.00434776	−0.00004795
160	18.47260446	0.00545794	2.02060394	−0.00011246
320	18.47260446	0.00269819	2.02281302	−0.00006155
640	18.47260446	0.00136043	1.98333827	0.00002267
1280	18.47260446	0.00070582	1.92745730	0.00005120

The Black-Scholes adjustment performs better than the smoothing terminal payoff procedure at low values of n.

Lookback options - Neumann boundary condition

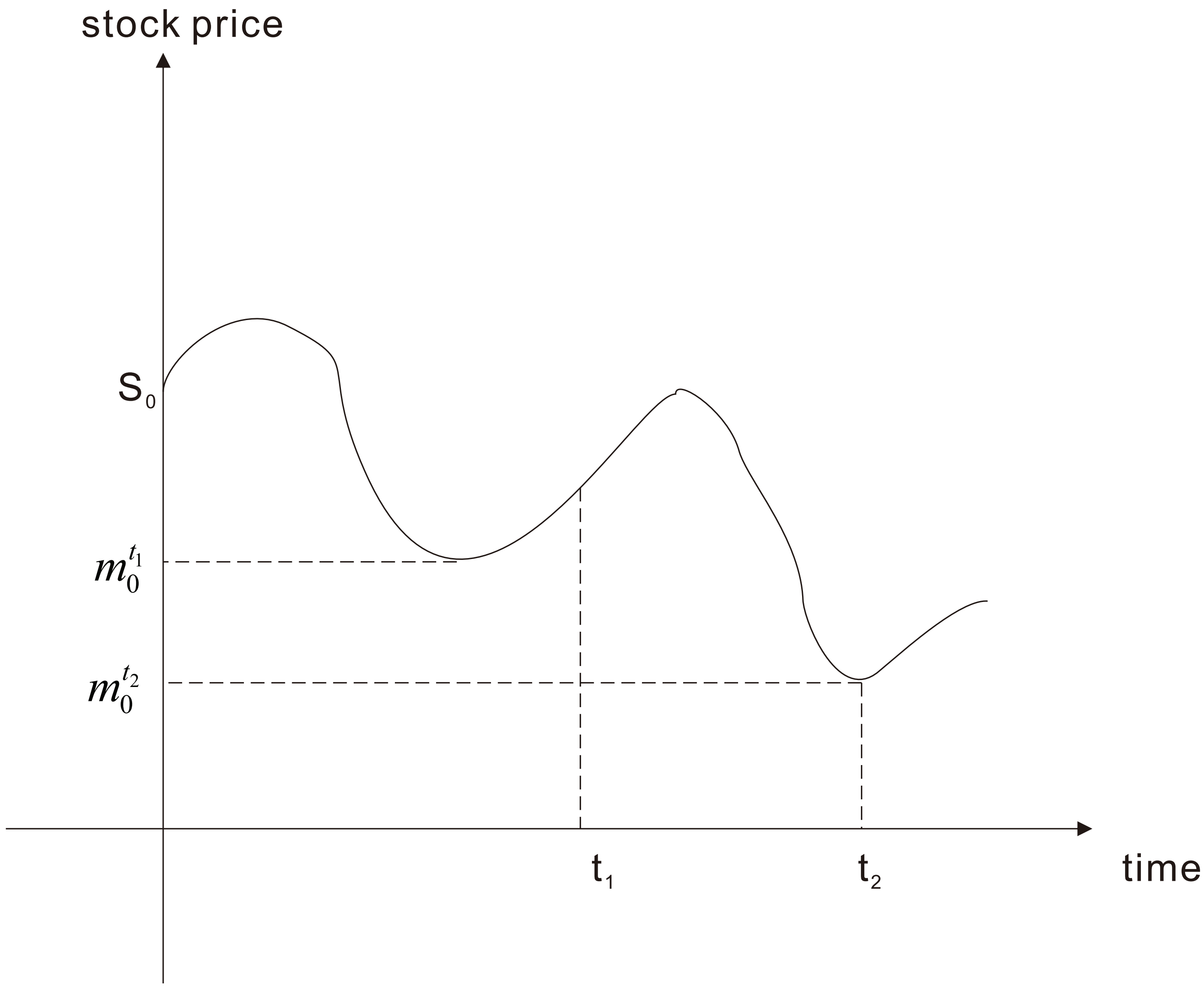
For the floating strike lookback option, by applying appropriate choices of similarity variables, the pricing formulation reduces to the form similar to that of usual one-asset option models, except that the Neumann boundary condition appears at the boundary of the semi-infinite domain of the lookback 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, $T_0 < t$. The terminal payoff at time T of the lookback call is given by

$$c(S, m, T) = S - m.$$

Note that $S \geq m$, so there is no optionality in the terminal payoff. The boundary condition at $S = m$ will be shown to be

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

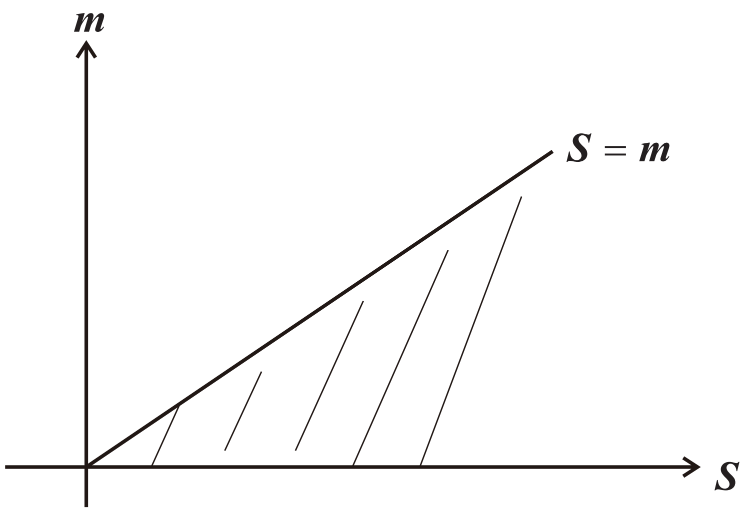


Realization of the continuously monitored minimum value of the stock price as the calendar time evolves.

At time t_1, the future distribution of $m_{t_1}^t$, $t > t_1$, depends on S_{t_1} and $m_0^{t_1}$. One has to ensure that the stock price goes below $m_0^{t_1}$ in order that an updated realized minimum value of stock price for $t > t_1$ is recorded.

However, at time t_2 at which $S_{t_2} = m_0^{t_2}$, the future distribution of $m_{t_2}^t$, $t > t_2$, depends on S_{t_2} only. This is somewhat like the situation at time zero where the recorded minimum value is simply S_0 so that m_0^t, $t > 0$, depends on S_0 only.

It is necessary to prescribe the boundary condition of the pricing model along $S = m$.



1. How to justify the boundary condition at $S = m$, where

$$\left.\frac{\partial c}{\partial m}\right|_{S=m} = 0?$$

Recall the terminal payoff depends on m_0^T, where $m_0^T = \min(m_0^t, m_t^T)$, $t < T$. When $S_t = m_0^t$, the future updating of the realized minimum value does not require the information of the current realized minimum value m_0^t (knowledge of S_t is sufficient). Hence, the call value $c(S_t, m_0^t, t)$ is independent of the current realized minimum value m_0^t.

2. Why the partial differential equation for the call value does not contain terms involving m?

The contribution to the differential change in call value due to differential change in m is given by $\frac{\partial c}{\partial m} dm$. We expect that this term should be added in the differential equation. However, we argue that either d$m = 0$ or $\frac{\partial c}{\partial m} = 0$ under two scenarios.

(i) When $S > m, dm = 0$. This is because the differential change of m would not occur in the next differential time interval dt since S is above m by a finite amount.

(ii) When $S = m, \frac{\partial c}{\partial m} = 0$ (as explained in the above).

Combining the above results, we conclude that $\frac{\partial c}{\partial m} dm = 0$.

For the *floating strike lookback call option*, the terminal payoff takes the form: $c(S, m, 0) = Sf\left(\frac{m}{S}\right)$.

Governing differential equation:

$$\frac{\partial c}{\partial \tau} = \frac{\sigma^2}{2} S^2 \frac{\partial^2 c}{\partial S^2} + (r - q)S\frac{\partial c}{\partial S} - rc, \quad S \geq m, \quad \tau > 0, \quad \tau = T - t$$

with auxiliary conditions:

$$\left.\frac{\partial c}{\partial m}\right|_{S=m} = 0 \quad \text{and} \quad c(S, m, 0) = S - m.$$

Here, m is a parameter that appears in the auxiliary conditions only. Note that $c(S, m, 0) = S\left(1 - \frac{m}{S}\right)$. This motivates us to choose the following set of similarity variables:

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

where $\tau = T - t$. The reduction of dimensionality of the option model can be achieved via the use of S as the numeraire.

These choices of transformation of variables are related to the use of share measure and the absorption of the discount term (the discount rate becomes q since stock price instead of money market account is used as the numerarie) together with the use of $\ln S$ as the independent state variable.

The drift rate of $\ln S$ under the share measure is $r - q - \frac{\sigma^2}{2} + \sigma^2 = r - q + \frac{\sigma^2}{2}$. Putting all these relations together, the Black-Scholes equation for $c(S, m, t)$ is transformed into the following equation for V:

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

Note that $S > m$ corresponds to $x > 0$. The discount term disappears since we define V as the undiscounted normalized call price via multiplying by the growth factor $e^{q\tau}$.

The terminal payoff condition becomes the following initial condition:

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

Neumann (reflecting) boundary condition at $x = 0$

Recall the boundary condition for $c(S, m, t)$ at $S = m$, where $\left.\frac{\partial c}{\partial m}\right|_{S=m} = 0$. Note that $\frac{\partial}{\partial m} = -\frac{1}{m}\frac{\partial}{\partial x}$ so that

$$\frac{\partial}{\partial m}\left(e^{-q\tau} VS\right) = e^{-q\tau} S\frac{\partial V}{\partial m} = -e^{-q\tau}\frac{S}{m}\frac{\partial V}{\partial x} = -e^{-q\tau}e^{x}\frac{\partial V}{\partial x}.$$

The boundary condition at $S = m$ or $x = 0$ becomes the Neumann (reflecting) condition:

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

Continuously monitored floating strike lookback call option

Using the explicit FTCS scheme, we obtain

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

where $\mu=\left(r-q+\frac{\sigma^2}{2}\right)\frac{\Delta\tau}{\Delta x}$ and $\alpha=\sigma^2\frac{\Delta\tau}{\Delta x^2}$.

For the continuously monitored lookback option model, 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

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

where V_{-1}^n is the option value at a fictitious node one cell to the left of node $j = 0$.

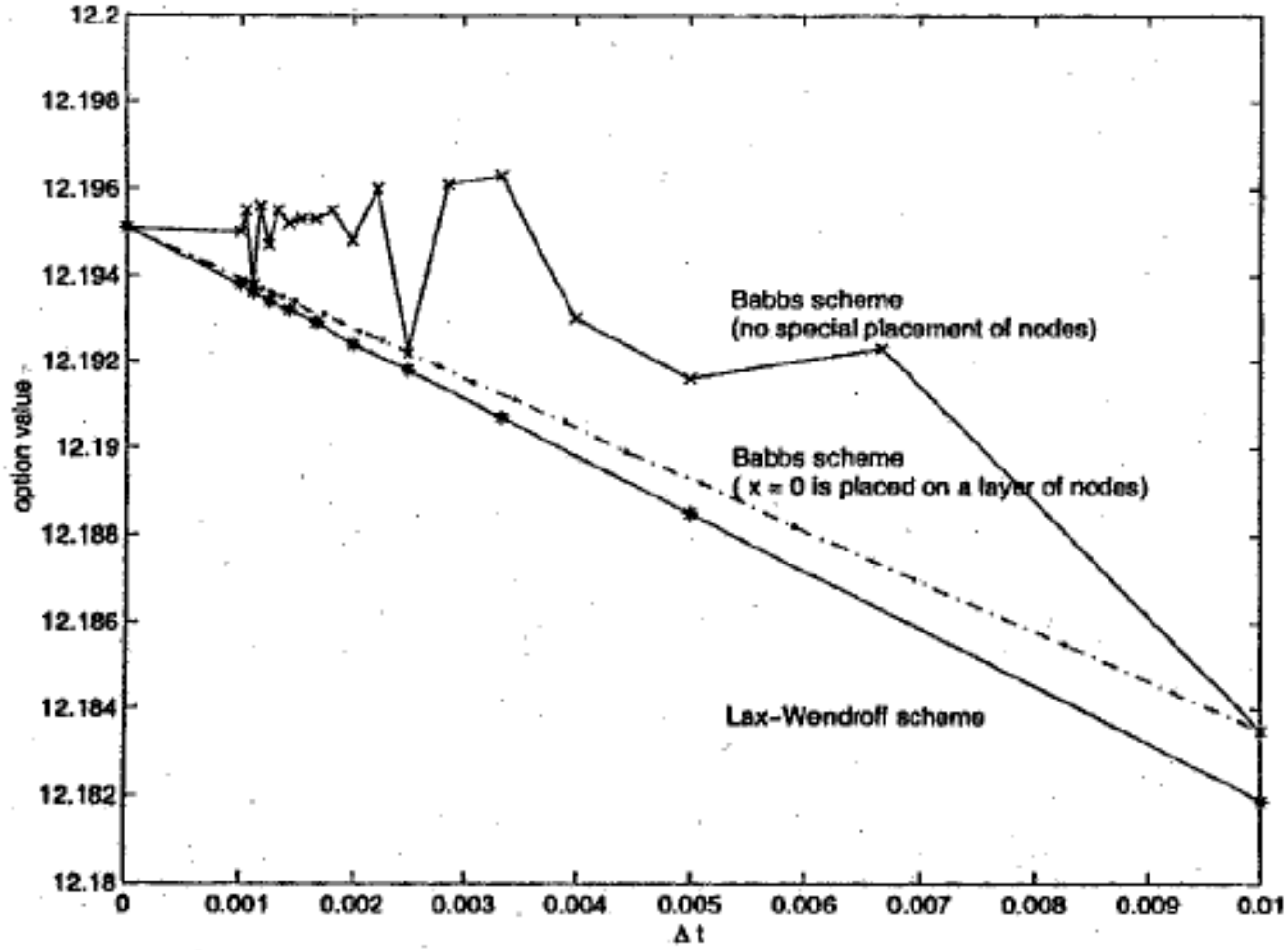
By setting $j = 0$ and applying the approximation of the Neumann condition: $V_1^n = V_{-1}^n$, we obtain

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

Numerical results obtained from the above FTCS scheme demonstrate $O(\Delta t)$ rate of convergence.

Remark

The Cheuk-Vorst scheme achieves only $O(\sqrt{\Delta t})$ convergence. This is due to the improper treatment of the numerical boundary condition at $S = m$ using pure probabilistic argument.



The convergence trend of the numerical continuously monitored lookback call option values obtained using the Babbs scheme depends sensibly on the positioning of the reflecting boundary. Linear rate of convergence is exhibited only when the reflecting boundary is placed on a layer of nodes. The parameter values used in the lookback option model are: $S = 100, m_{T_0}^t = 100e^{-0.1}, r = 4\%, q = 2\%, \sigma = 10\%, \tau = 1.0$.

Discretely monitored floating strike lookback call option

The realized minimum of the asset price is updated only on a discrete set of monitoring instants.

The Black-Scholes equation remains valid between two successive monitoring dates. It is possible to have $S < m$ since no updating of m is recorded at time not falling on a monitoring date. The Neumann boundary condition $\left.\frac{\partial V}{\partial x}\right|_{x=0} = 0$ should not be applied at time steps that do not correspond to a monitoring instant. For numerical calculations, the usual finite difference calculations are performed as that of a vanilla option at those time levels not corresponding to a monitoring instant. The domain of x would be $(-\infty, \infty)$ instead of $(0, \infty)$.

Suppose the n^{th} time level happens to be a monitoring instant, we set the numerical option values to the left of $x = 0$ to be

$$V_j^n = V_0^n, \quad j = -1, -2, \cdots .$$

Effectively, we can save the effort to compute V_j^n, $j = -1, -2, \ldots$.

To explain the above property by intuitive argument, we consider the time right before a monitoring instant. Provided that $S < m$, the lookback option value is the same as that of $S = m$ since an updated minimum value will be recorded immediately after the monitoring instant; so $\dfrac{\partial V}{\partial x}(x, \tau) = 0$, $x \leq 0$.

Positioning of $x = 0$

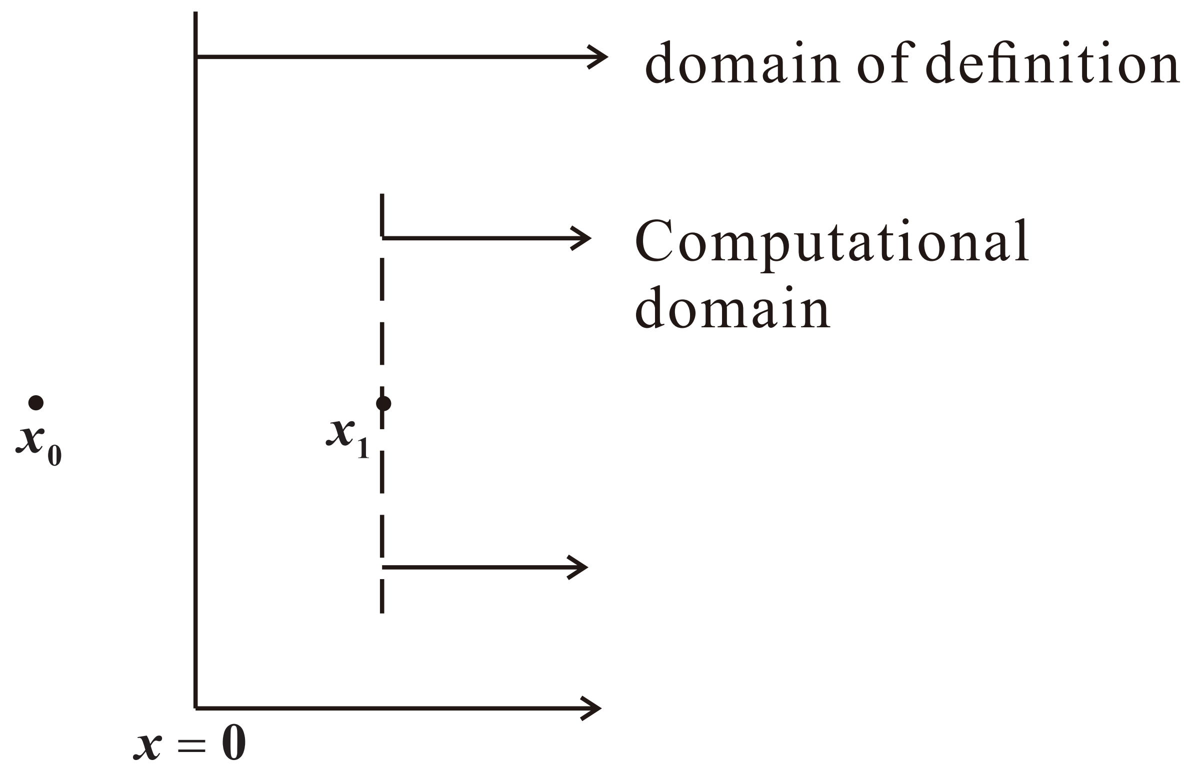
We may either place the boundary $x = 0$ at x_0 or between the two nodes x_0 and x_1. Suppose $x = 0$ is placed between x_0 and x_1 at the n^{th} time step which coincides with a monitoring instant, we then have

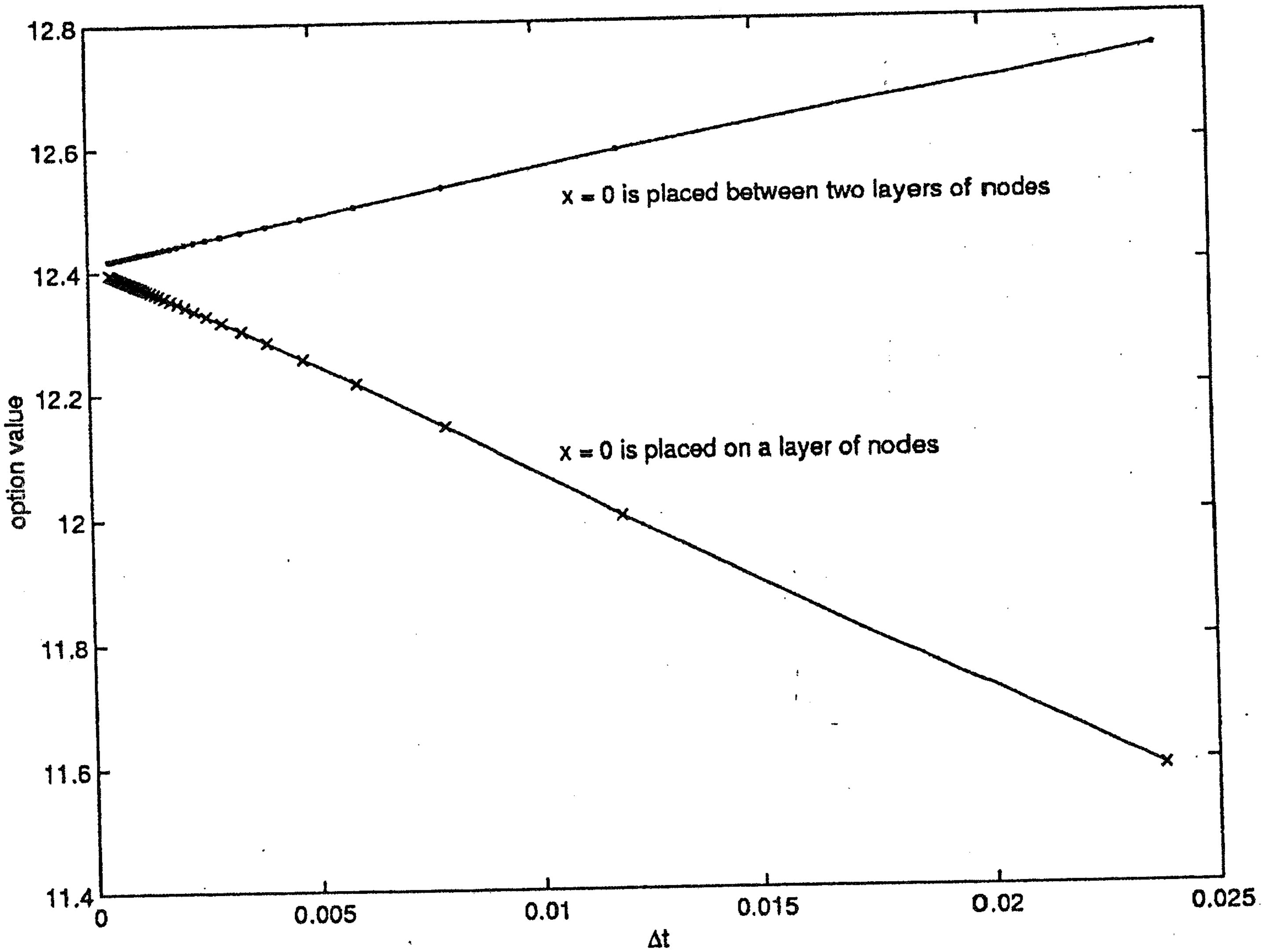
$$V_1^n = V_0^n = V_{-1}^n = V_{-2}^n = \ldots$$

By substituting $V_1^n = V_0^n$ in the explicit FTCS scheme at $j = 1$ on a monitoring instant, we obtain an alternative form for the scheme at the boundary node x_1 as follows:

$$V_1^{n+1} = \frac{\alpha + \mu}{2} V_2^n + \left(1 - \frac{\alpha + \mu}{2}\right) V_1^n.$$

At a time step that corresponds to a monitoring instant, x_1 is a boundary node along the boundary of the computational domain while x_0 is a fictitious node outside the computational domain. The domain of definition is $x \geq 0$. The left boundary of the computational domain is placed at $x = \dfrac{\Delta x}{2}$, which is not overlapping exactly with $x = 0$ as in the continuously monitored case.

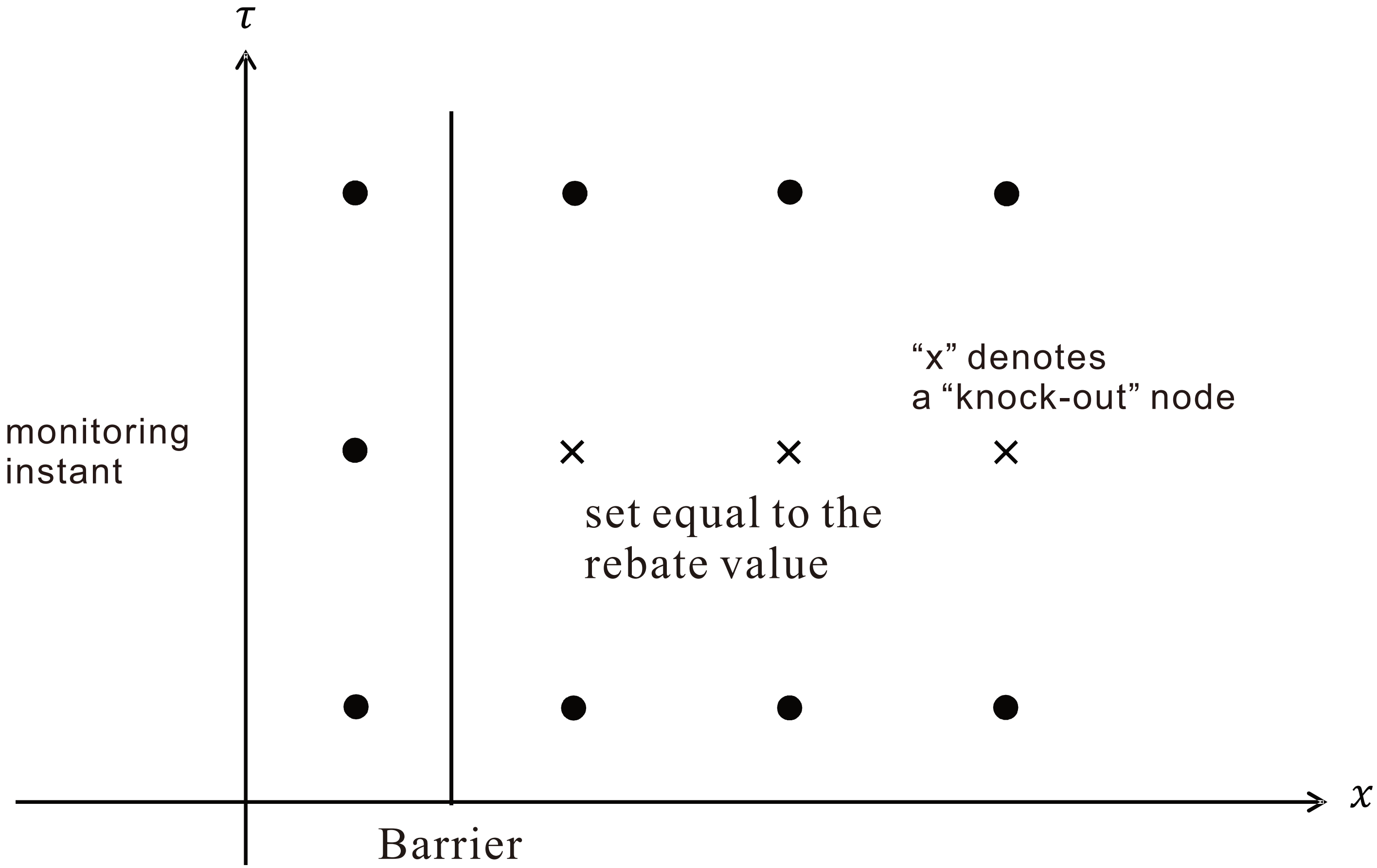




The figure shows the plots of numerical option value against time step Δt for a discretely monitored floating strike lookback call option.

Discretely monitored barrier options

For an one-sided up-and-out discretely monitored barrier option, the option values are set to be the rebate value at nodes that lie in the knock-out region at those time steps that correspond to monitoring instants.



Remarks

- The computational domain is taken to be the same as that of a vanilla option. We enforce the knock-out condition only on those time instants that correspond to the discrete monitoring instants in the option contract.

- Perform the usual explicit scheme calculations at all nodes, noting that the numerical option values at the "knock-out" nodes are set to be the rebate value.

- The barrier is placed mid-way between two vertical layers of nodes. This is in contrast to the continuously monitored barrier option model where the knock-out barrier is placed along a vertical layer of nodes.

Down-barrier proportional step call option

The terminal payoff of the *down-barrier proportional-step call option* is defined to be

$$\exp(-\rho\tau_B^-)\max(S_T - K, 0),$$

where ρ is called the penalty rate and τ_B^- is the occupational time in the down knock-out region defined by

$$\tau_B^- = \int_{t_0}^{T} H(B - S_t)\,dt,$$

where the step function is defined by

$$H(B - S_t) = \begin{cases} 1 & S_t \le B \\ 0 & \text{otherwise} \end{cases}.$$

To capture this penalty rate feature in the discrete numerical scheme, we define the damping factor over one time step to be

$$d_j = \begin{cases} e^{-\rho\Delta t} & \text{if } x_j \le \ln B \\ 1 & \text{otherwise} \end{cases}.$$

We accumulate the additional discounting effect on the terminal payoff over successive time steps whenever the stock price stays below the down-barrier B, and no damping effect if otherwise. The explicit FTCS scheme is given by

$$V_j^{n+1} = \left[\frac{\alpha + \mu}{2} V_{j+1}^n d_{j+1} + (1-\alpha) V_j^n d_j + \frac{\alpha - \mu}{2} V_{j-1}^n d_{j-1}\right] e^{-r\Delta t},$$

where $\mu = \left(r - q - \frac{\sigma^2}{2}\right) \frac{\Delta \tau}{\Delta x}$ and $\alpha = \sigma^2 \frac{\Delta \tau}{\Delta x^2}$. Note that the penalty discount factor should be applied based on the positions x_{j+1}, x_j and x_{j-1} in the next time instant (n^{th} time level). Therefore, we multiply the nodal values by the respective penalty discount factor.

How do we incorporate the damping effect in the knock-out region into the implicit Crank-Nicolson scheme and fully implicit scheme? The discount term in the governing partial differential equation becomes $-[r + \rho H(B - S)]V$ instead of $-rV$. We simply discretize the new discount term accordingly.

Summary of the optimal choices of the positioning of the nodes relative to the barrier to achieve smooth linear rate of convergence of the numerical solutions at asymptotically zero value of the time step for different types of the path-dependent options and monitoring features

		Barrier placed between two layers of nodes	Barrier on a layer of nodes
Barrier options	continuously monitored		✓
	discretely monitored	✓	
Lookback options	continuously monitored		✓
	discretely monitored	✓	✓
Proportional-step options	continuously monitored	✓	

3.3 Properties of numerical finite difference solutions

Truncation errors and order of convergence

The local truncation error of a given finite difference 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. The local truncation error at the node point $(j\Delta x, n\Delta\tau)$ of the explict FTCS scheme is given by substituting the exact solution into the explicit scheme:

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

We then expand each term by performing the Taylor expansion at the node point $(j\Delta x, n\Delta\tau)$.

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

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

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

The leading terms in $T(j\Delta x, n\Delta\tau)$ are $O(\Delta\tau)$ and $O(\Delta x^2)$.

- 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 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.

- 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, $\Delta x^2 = \lambda^2\Delta\tau$ for some finite constant λ, then the leading truncation error terms 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 expected since the CRR binomial scheme is a special case of the FTCS scheme, and all binomial schemes have the same order of accuracy.

- For the implicit Crank-Nicolson scheme, it is second order time accurate and second order space accurate. However, if we choose $\Delta\tau \sim \Delta x^2$, then the advantage of being second order time accurate disappears since the scheme becomes essentially second order space accurate [which is $O(\Delta\tau)$].

- 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 Problem 6.20 in Kwok's text).

Extrapolation techniques

The numerical solution $V_j^n(\Delta\tau)$ using time step $\Delta\tau$ has the asymptotic expansion of the form

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

where $V_j^n(0)$ is visualized as the exact solution of the continuous model 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 $\dfrac{\Delta\tau}{2}$ successively,

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

Hence, $V_j^n(0)$ can be estimated using $V_j^n(\Delta\tau)$ and $V_j^n\left(\dfrac{\Delta\tau}{2}\right)$ via

$$V_j^n(0) \approx \frac{2^m V_j^n\left(\frac{\Delta\tau}{2}\right) - V_j^n(\Delta\tau)}{2^m - 1}.$$

When $m = 1, V_j^n(0)$ can be estimated from extrapolation as

$$V_j^n\left(\frac{\Delta\tau}{2}\right) + \left[V_j^n\left(\frac{\Delta\tau}{2}\right) - V_j^n(\Delta\tau)\right].$$

Numerical stability of finite difference schemes

- Consistency is only a necessary but not sufficient condition for convergence of the numerical solution to the solution of the continuous model.

- The roundoff errors incurred during numerical calculations may lead to the blow up of the solution and erode the whole computation.

- 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 numerical stability is the necessary and sufficient condition for the convergence of a consistent finite difference scheme.

Example – Erosion of numerical calculations by roundoff errors

Consider the evaluation of $I_n = \int_0^1 \frac{x^n}{x+5}\,dx, n = 0, 1, 2, \cdots, 20;$ using the property: $I_n + 5I_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5}\,dx = \int_0^1 x^{n-1}\,dx = \left.\frac{x^n}{n}\right]_0^1 = \frac{1}{n},$ and $\int_0^1 \frac{1}{x+5}\,dx = \ln|x+5|\Big|_0^1 = \ln\frac{6}{5}$, we deduce the following relation:

$$I_n + 5I_{n-1} = \frac{1}{n}, n = 1, \cdots, 20; I_0 = \ln\frac{6}{5}.$$

Since $I_n < I_{n-1}$ and $I_n > 0$, so $5I_{n-1} < \underbrace{I_n + 5I_{n-1}}_{\frac{1}{n}} < 6I_{n-1}$. We then have

$$\frac{1}{6n} < I_{n-1} < \frac{1}{5n}.$$

Forward Iteration: Starting with $I_0 = \ln\frac{6}{5}$, compute

$$I_1 = \frac{1}{1} - 5I_0, \quad I_2 = \frac{1}{2} - 5I_1, \text{ etc.}$$

n	Column A In Forward Iteration	Column B In Backward Iteration	n	Column A In Forward Iteration	Column B In Backward Iteration
0	0.18232155	0.18232155	11	0.017324710	0.014071338
1	0.088392216	0.088392216	12	−0.003290219	0.012976641
2	0.058038918	0.058038919	13	−0.093374172	0.012039867
3	0.043138742	0.043138734	14	−0.39544229	0.011229233
4	0.034306287	0.034306329	15	2.0438781	0.010520499
5	0.028468560	0.028468352	16	−10.156890	0.009897504
6	0.024323864	0.024324995	17	50.843276	0.009336007
7	0.021237820	0.021232615	18	−254.16082	0.008875522
8	0.018810897	0.018836924	19	1270.8567	0.0082539682
9	0.017056624	0.016926489	20	−6354.2338	0.0087301587
10	0.014716876	0.015367550			

Implementation of the Forward Iteration Calculations on a computer with 8 significant figures leads to the results tabulated in Column A. The successive numerical values alternate sign and increase in magnitude.

Propagation of roundoff error

Exact relation: $I_1 = -5I_0 + 1$. Taking an approximate initial value $\widehat{I}_0$, the calculated value of the first iterate $\widehat{I}_1 = -5\widehat{I}_0 + 1$.

Here, we assume no further errors subsequent calculations except that $I_0 = \ln 6/5$ cannot be represented exactly on a computer. Note that

$$I_1 - \widehat{I}_1 = (-5)(I_0 - \widehat{I}_0),$$

so that the initial error $I_0 - \widehat{I}_0$ is magnified by a factor of -5 after each iteration. Deductively,

$$I_n - \widehat{I}_n = (-5)^n(I_0 - \widehat{I}_0).$$

Backward iteration: Taking $I_{20} \approx \frac{1}{2}\left(\frac{1}{6 \times 21} + \frac{1}{5 \times 21}\right) = 0.0087301587.$

Implementation: $I_{n-1} = -\frac{I_n}{5} + \frac{1}{5n}, n = 20, 19, \cdots, 1; I_{20} = 0.0087301587.$

We obtain $I_0 - \widehat{I}_0 = \left(-\frac{1}{5}\right)^n (I_n - \widehat{I}_n)$ (see the results shown in column B).

Fourier method of stability analysis

The Fourier method is based on the assumption that the numerical scheme admits a solution of the form

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

where k is the wavenumber and $i = \sqrt{-1}$. This resembles the method of separation of variables in solving partial differential equations, where V is decomposed as $V(x,t) = X(x)T(t)$.

- Here, $e^{ikj\Delta x} = e^{ikx}\Big|_{x=j\Delta x}$ represents the Fourier mode with wavenumber k, $A^n(k)$ represents the amplitude of the k^{th} mode at the n^{th} time level.

  The nodal values are related by

$$\begin{aligned} V_{j+1}^{n+1} &= \frac{A^{n+1}(k)}{A^n(k)} e^{ik\Delta x} A^n(k) e^{ikj\Delta x} = \frac{A^{n+1}(k)}{A^n(k)} e^{ik\Delta x} V_j^n, \\ V_{j-1}^{n+1} &= \frac{A^{n+1}(k)}{A^n(k)} e^{-ik\Delta x} V_j^n, \quad V_{j+1}^n = e^{ik\Delta x} V_j^n, \text{etc.} \end{aligned}$$

- The *von Neumann stability criterion* examines the growth of the above Fourier component.

Substituting the Fourier representation of the solution into the two-level six-point scheme:

$$a_1 V_{j+1}^{n+1} + a_0 V_j^{n+1} + a_{-1} V_{j-1}^{n+1} = b_1 V_{j+1}^{n} + b_0 V_j^{n} + b_{-1} V_{j-1}^{n},$$

we obtain

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

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

$$|G(k)| \leq 1,$$

for $0 \leq k\Delta x \leq \pi$. This is because the larger wavenumber of the sinusoidal wave that can be resolved by the grids is limited by $\pi/\Delta x$. Henceforth, we write $\beta = k\Delta x$.

Stability of the Cox-Ross-Rubinstein binomial scheme

Consider the binomial scheme

$$V_j^{n+1} = [pV_{j+1}^n + (1-p)V_{j-1}^n]e^{-r\Delta\tau},$$

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

$$\begin{aligned} G(\beta) &= [pe^{i\beta} + (1-p)e^{-i\beta}]e^{-r\Delta\tau} \\ &= [p(\cos\beta + i\sin\beta) + (1-p)(\cos\beta - i\sin\beta)]e^{-r\Delta\tau} \\ &= (\cos\beta + iq\sin\beta)e^{-r\Delta\tau}, \quad q = 2p-1. \end{aligned}$$

The von Neumann stability condition requires

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

When $0 \leq p \leq 1$, we have $|q| \leq 1$ so that $|G(\beta)| \leq 1$ for all β. Under this reasonable condition on the probability of up-move, the scheme is guaranteed to be stable in the von Neumann sense.

- Sufficient condition for von Neumann stability of the Cox-Ross-Rubinstein scheme: non-occurrence of negative probability values in the binomial scheme. This condition coincides with the intuition that probabilities of up-jump and down-jump cannot be negative.

- Later, we will show that non-negativity of coefficients in the numerical scheme is also necessary to avoid spurious oscillations in numerical calculations.

Stability of the Crank-Nicolson scheme

The corresponding amplification factor of the Crank-Nicolson scheme is

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

The von Neumann stability condition requires

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

$$0 \leq \beta \leq \pi.$$

The above condition is satisfied for any choices of $\Delta\tau$ and Δx. Hence, the Crank-Nicolson scheme is unconditionally stable.

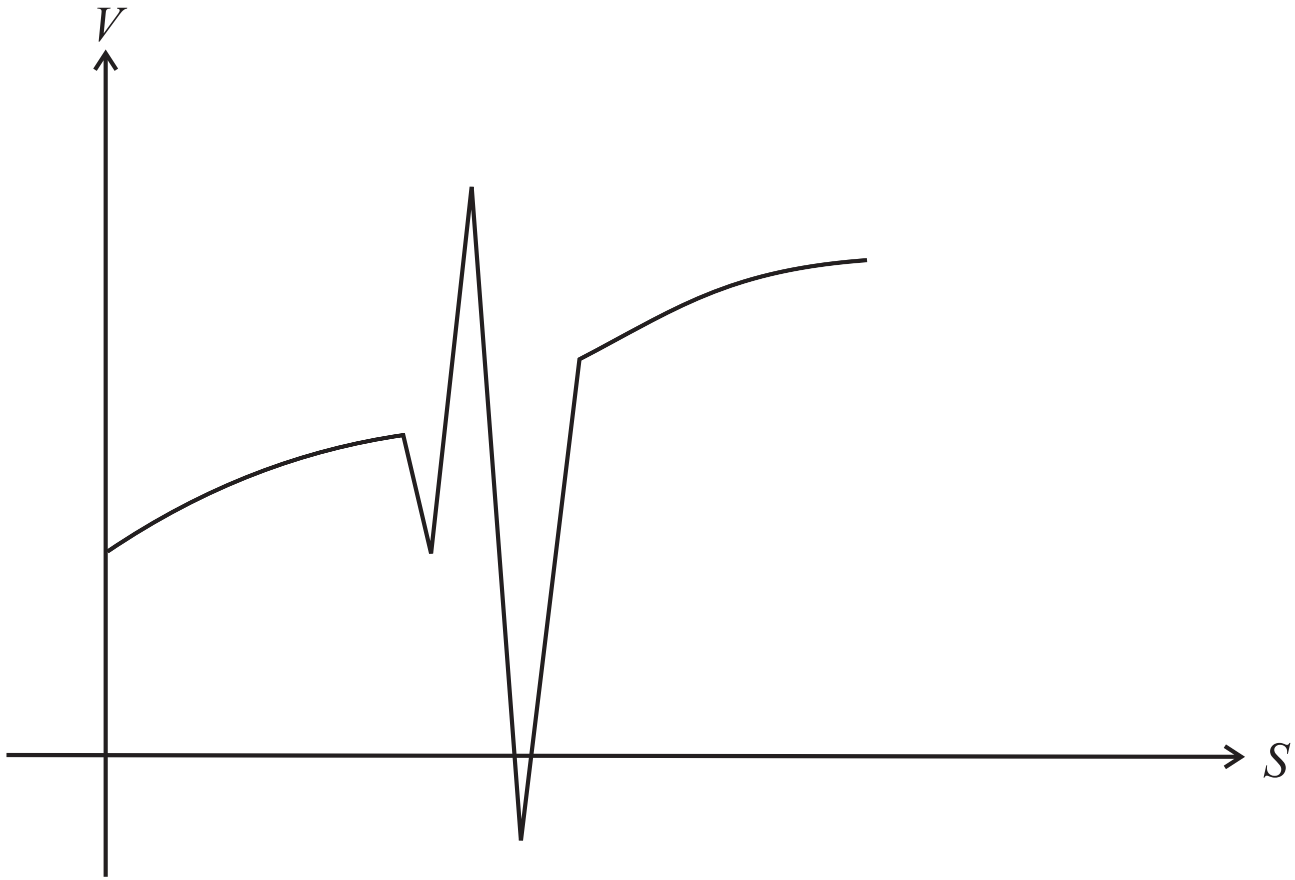
Order of accuracy and stability of the Crank-Nicolson scheme

- 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 Crank-Nicholson solution can be much deteriorated due to non-smooth property of the terminal payoff function in most option models. Also, if we maintain $O(\Delta\tau) = O(\Delta x^2)$, then the overall accuracy is still $O(\Delta x^2)$, same as that of the FTCS scheme.

Spurious oscillations



Spurious oscillations in numerical solution of an option price.

Another undesirable feature in the behavior 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.

Boundedness of numerical solution and non-negative coefficients

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

$$\max_j |V_j^n| \leq M \quad \text{for all} \quad n \geq 1.$$

Proof

Consider the two-level explicit scheme

$$V_j^{n+1} = b_{-1} V_{j-1}^n + b_0 V_j^n + b_1 V_{j+1}^n,$$

we deduce that

$$|V_j^{n+1}| \leq |b_{-1}|\,|V_{j-1}^n| + |b_0|\,|V_j^n| + |b_1|\,|V_{j+1}^n|,$$

Since b_{-1}, b_0 and b_1 are non-negative, the inequality persists when we take maximum among nodal values at the same time level, so

$$\max_j |V_j^{n+1}| \leq b_{-1} \max_j |V_{j-1}^n| + b_0 \max_j |V_j^n| + b_1 \max_j |V_{j+1}^n|.$$

Let E^n denote $\max_j |V_j^n|$, the above inequality can be expressed as

$$E^{n+1} \leq b_{-1}E^n + b_0 E^n + b_1 E^n = E^n$$

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

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

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 $V_j^0 = 0, j \neq 0$. 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. Note that $|V_{-1}^1| + |V_0^1| + |V_1^1| = (b_{-1} - b_0 + b_1)\epsilon = (1 - 2b_0)\epsilon$.

This alternating sign property can be shown to persist at all later time levels. Sum of moduli of solution values $= (|b_1| + |b_0| + |b_{-1}|)\epsilon = (1 - 2b_0)\epsilon > \epsilon$. The solution values oscillate in signs at neighboring nodes. The oscillation amplitudes grow with an increasing number of time steps. Let $\mathcal{S}^n = \sum_j |V_j^n|$, then $\mathcal{S}^n = (1 - 2b_0)^n \mathcal{S}_0 = (1 - 2b_0)^n \epsilon$.

Summary - Considerations in the construction of finite difference schemes for pricing options

- Derivation of the model formulation – governing partial differential equation plus auxiliary conditions.
- Discretization of the equation – approximate the continuous differential / integral operators by appropriate difference operators.
- Choice of the computational domain
  - truncation of the infinite / semi-infinite domain of the continuous model
  - placement of the knock-out barrier (different choices of the boundary of the computational domain with respect to continuous versus discrete monitoring)

- Approximation of the auxiliary conditions
  - Initial conditions are derived from the terminal payoff function.
  - Boundary conditions are derived either from the contractual specification (say, rebate value after knock-out) or properties of the price function (says, zero gamma at high stock price level).
  - When the boundary of the computational domain is still within the domain of definition of the continuous model, one may adopt a skew computational stencil (avoidance of nodal points outside the computational domain) at the boundary nodes on the assumption that the option values at the boundary nodes remain to satisfy the governing equation.

- Sources of discretization errors
  - Difference approximation to the differential operators.
  - Numerical approximation of the auxiliary conditions in the option models.

- To avoid spurious oscillations in the numerical solution values, the coefficients in the explicit finite difference schemes must be all positive.

- Numerical stability considerations – accuracy of numerical calculations should not be eroded by the accumulation of roundoff errors.

- Implicit schemes versus explicit schemes

  - Most implicit schemes are unconditionally stable while explicit schemes normally require some stringent time step restriction in order to maintain numerical stability.
  - Implementation of implicit schemes can be performed effectively by the Thomas algorithm (solution by Gaussian elimination of the associated tridiagonal system of equations).
  - For pricing an American option with the early exercise feature, the explicit scheme uses the straightforward dynamic programming procedure while the implicit scheme has to adopt the projected successive-over-relaxation method.