Advanced Numerical Methods

Topic 2 – Finite difference methods

2.1 Discretization of the Black-Scholes equation

- Explicit schemes
- Crank-Nicolson scheme

2.2 Pricing of American options

- Front fixing method
- Point relaxation method

2.3 Numerical approximation of auxiliary conditions

- Barrier options
- Lookback options

2.4 Properties of numerical solutions

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

2.1 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: $\tau = T - t, x = \ln S,$

$$\frac{\partial}{\partial t} = -\frac{\partial}{\partial \tau}, \quad \frac{\partial}{\partial S} = \frac{1}{S}\frac{\partial}{\partial x} \quad \text{or} \quad S\frac{\partial}{\partial S} = \frac{\partial}{\partial x}$$

$$\frac{\partial^2}{\partial x^2} = S\frac{\partial}{\partial S}\left(S\frac{\partial}{\partial S}\right) = S^2\frac{\partial^2}{\partial S^2} + S\frac{\partial}{\partial S} \text{ so that } S^2\frac{\partial^2}{\partial S^2} = \frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial x}.$$

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 time step 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. Note that proportional jumps on the asset price are adopted in the binomial/trinomial tree calculations.

Discretization of the domain

Transform the domain of the continuous problem

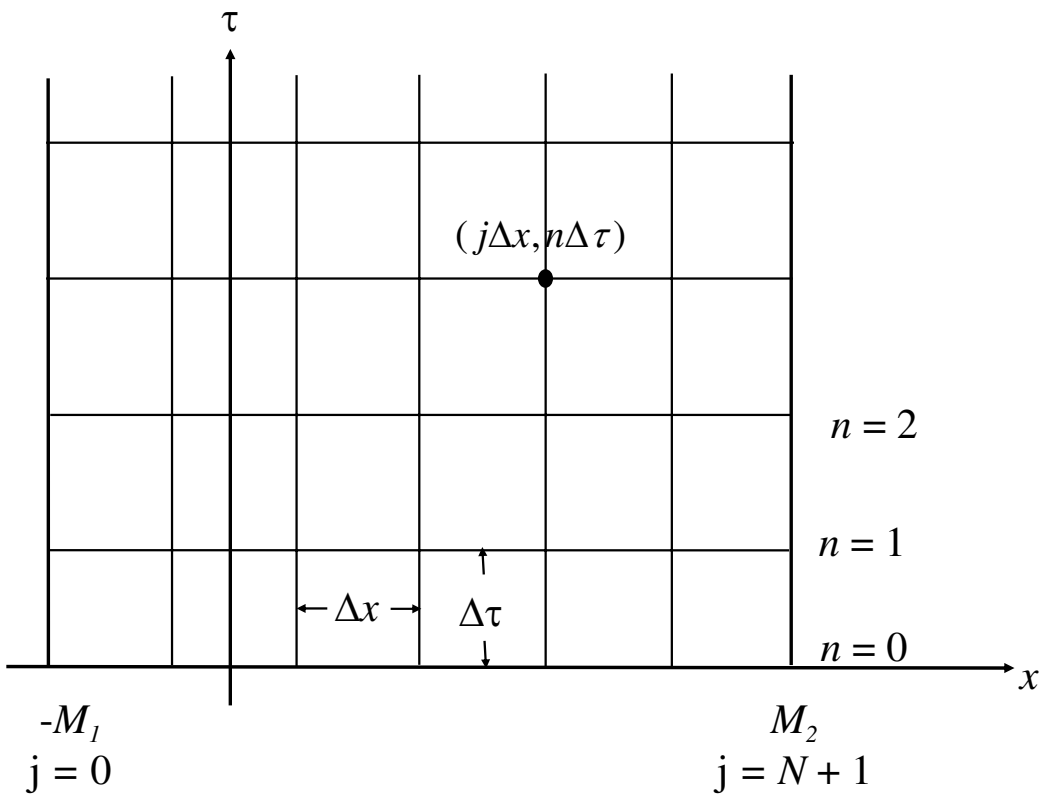
$$\{(x,\tau) : -\infty < x < \infty, \quad \tau \geq 0\}$$

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 $\Delta\tau = O(\Delta x^2)$.

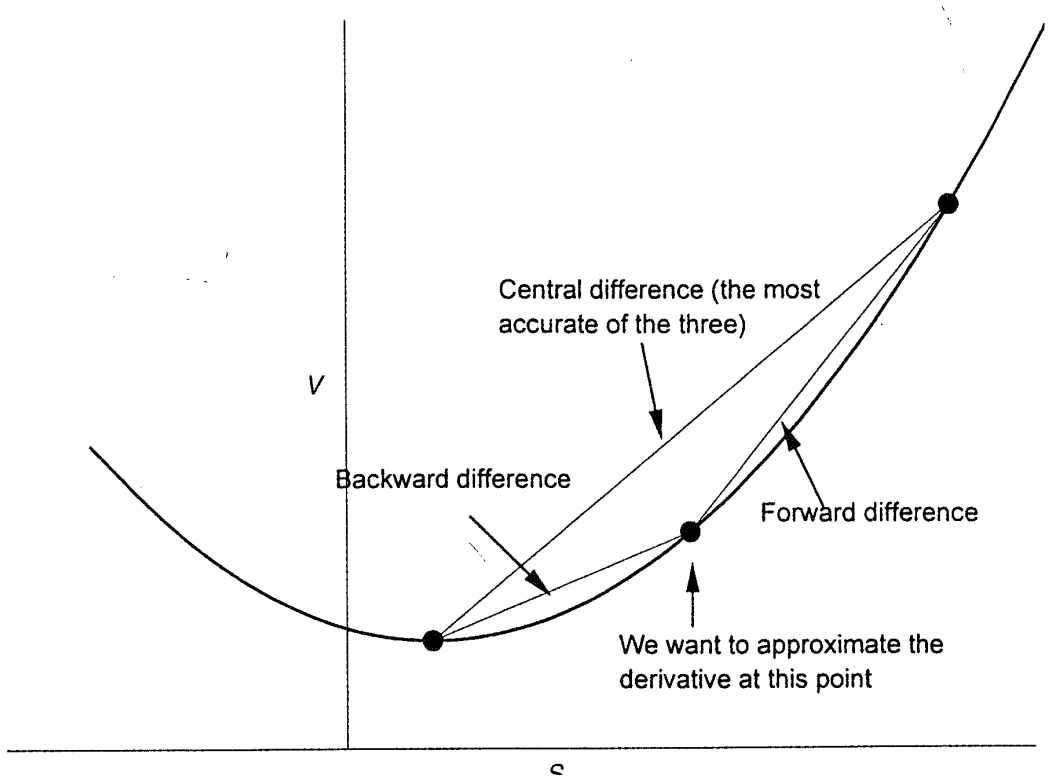
While we perform backward induction in trinomial calculations (going backwards in calendar time), we march forwards in the temporal variable τ (time to expiry) in the finite difference calculations.



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 $(j\Delta x, n\Delta\tau)$ 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 $\frac{\partial V}{\partial S}$.

Why the centered difference achieves higher order of accuracy compared to the 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

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

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:

$$\begin{aligned} & \frac{f(x+\Delta x)-f(x)}{\Delta x} \\ = \; & f'(x)+\frac{f''(x)}{2}\Delta x+O(\Delta x)^2 \end{aligned}$$

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

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

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:

$$f''(x) \approx \frac{2f(x_0) - 5f(x+\Delta x) + 4f(x+2\Delta x) - f(x+3\Delta x)}{\Delta x^2} + O(\Delta x^2).$$

Similarly, the backward difference formula is given by

$$f''(x) \approx \frac{2f(x_0) - 5f(x-\Delta x) + 4f(x-2\Delta x) - f(x-3\Delta x)}{\Delta x^2} + O(\Delta x^2).$$

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 taking

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

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

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

- Suppose we are given "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 $\tau = \triangle\tau$.

- 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

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

Both the binomial and trinomial schemes are members of the family when the reconnecting 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, $\ln d = -\triangle x$. The binomial scheme can be expressed as

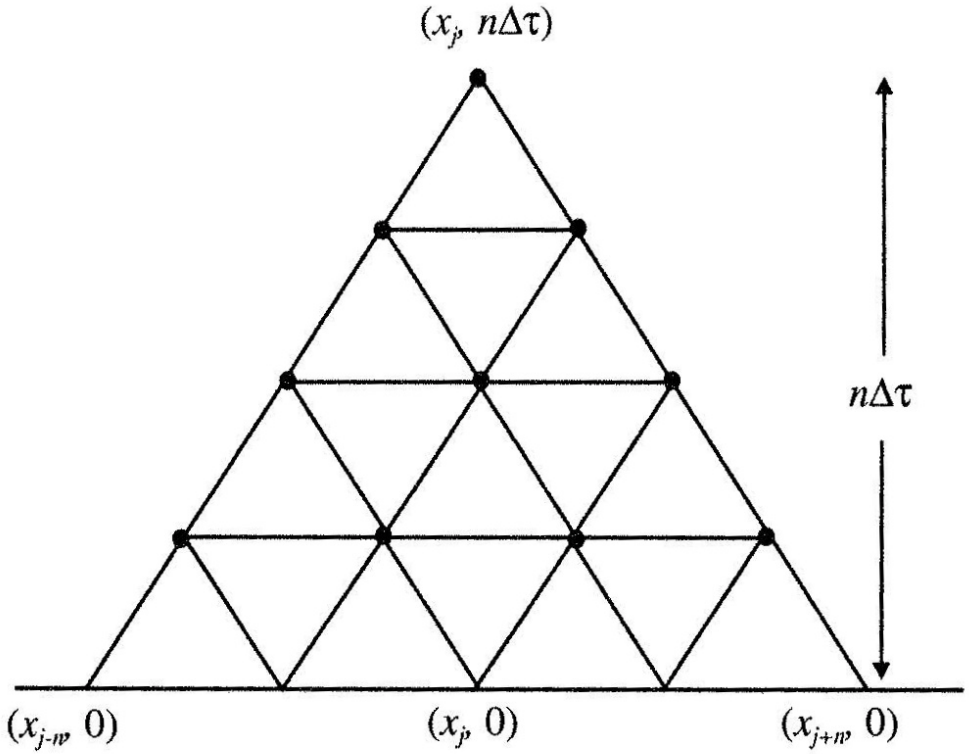
$$V^{n+1}(x) = \frac{pV^n(x + \triangle x) + (1-p)V^n(x - \triangle x)}{R}, \quad x = \ln S, \text{ and } R = e^{r\triangle\tau},$$

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, they are related by $\Delta x = \ln u = \sigma\sqrt{\Delta\tau}$ 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 $\lambda \geq 1$.

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.

In terms of x, the width of the domain of dependence $= 2n\Delta x = 2n \ln u = 2n\sigma\sqrt{\Delta\tau}$.

Negligence of boundary conditions

- With respect to $x = \ln S$, the width of the domain of dependence of a binomial scheme can be shown to be $\sqrt{n}$, 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.

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

- For example, an up-and-out put option with an upstream knock-out barrier B 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.

- 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 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 the barrier option models. To achieve a high level of numerical accuracy, it is important that the numerical scheme takes into account the effect of boundary conditions.

Final conditions and payoffs

At expiry the option value is just the payoff function, we have

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

or, in our finite-difference notation,

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

For example, if we are pricing a call option we put

$$V_j^0 = \max(j\,\Delta S - X, 0).$$

Boundary conditions

We must specify the option value along the two boundaries of the computational domain. What we specify will 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 $S - Xe^{-r(T-t)}$.

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

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

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

so that

$$V^n_{N+1} = \frac{5V^n_N - 4V^n_{N-1} + V^n_{N-2}}{2}.$$

We obtain the boundary value V^n_{N+1} in terms of interior values V^n_N, V^n_{N-1} and V^n_{N-2}.

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

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

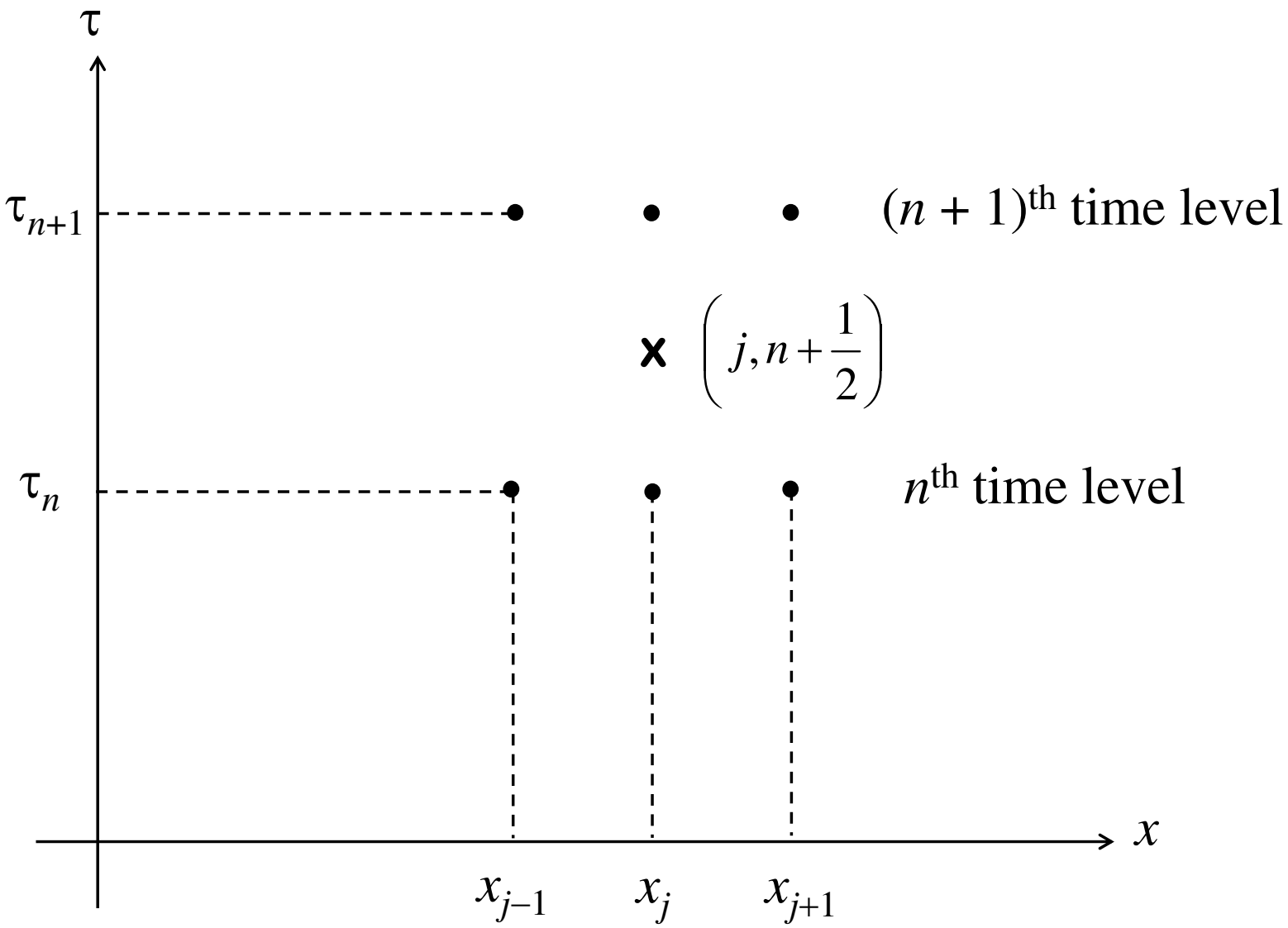
we then obtain the following two-level implicit finite difference scheme

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

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 of the form

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



The numerical scheme involves 3 option values at each of 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.

$$\begin{aligned}
\frac{\partial^2 V}{\partial x^2}\Big)_{j,n+\frac{1}{2}} &\approx \frac{1}{2}\left[\frac{\partial^2 V}{\partial x^2}\Big|_{j,n+1} + \frac{\partial^2 V}{\partial x^2}\Big|_{j,n}\right] \\
\frac{\partial V}{\partial \tau}\Big)_{j,n+\frac{1}{2}} &\approx \frac{V_j^{n+\frac{1}{2}+\frac{1}{2}} - V_j^{n+\frac{1}{2}-\frac{1}{2}}}{2\left(\frac{\Delta\tau}{2}\right)} = \frac{V_j^{n+1} - V_j^n}{\Delta\tau}.
\end{aligned}$$

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. The two-level six-point scheme can be represented as

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

where

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

Note that V_0^{n+1} and V_{N+1}^{n+1} are known values available from the boundary conditons. The solution of the above tridiagonal system can be effected by the well known *Thomas algorithm*.

Thomas algorithm

Consider the solution of the following tridiagonal system of the form

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

with $V_0 = V_{N+1} = 0$. This form is more general in the sense that the coefficients can differ among equations. Note 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 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.

- 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, namely

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

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 V_{k+1} and V_{k+2}, namely,

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

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

$$V_0 - e_0 V_1 = f_0,$$

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

$$e_0 = f_0 = 0.$$

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.

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

- 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.

- 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.

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: $I_n + 5I_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5}\,dx = \int_0^1 x^{n-1}\,dx = \frac{x^n}{n}\Big]_0^1 = \frac{1}{n}$, 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 $5I_{n-1} < \underbrace{I + 5I_{n-1}}_{\frac{1}{n}} < 5I_{n-1}$. 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 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 $I_{20} \approx \dfrac{1}{2}\left(\dfrac{1}{6 \times 21} + \dfrac{1}{5 \times 21}\right) = 0.0087301587.$

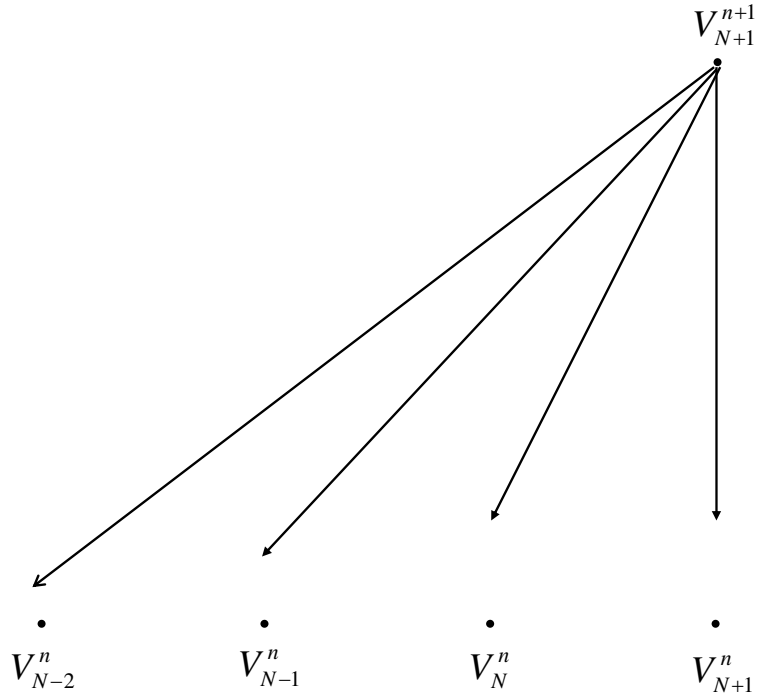
Implementation: $I_{n-1} = -\dfrac{I_n}{5} + \dfrac{1}{5n}, n = 20, 19, \cdots, 1; I_{20} = 0.0087301587.$

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

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 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.

- The option values at $j = N + 1$ are not prescribed by any boundary conditions arising from the continuous option model.

- It would create unnecessary errors if we arbitrarily set inappropriate boundary values.

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

We discretize the Black-Schole equation using one-sided backward difference:

$$\left.\frac{\partial V}{\partial x}\right|_{j=N+1} \approx \frac{V_{N-1} - 4V_N + 3V_{N+1}}{2\Delta x};$$
$$\left.\frac{\partial^2 V}{\partial x^2}\right|_{j=N+1} \approx \frac{V_{N+1} - 5V_N + 4V_{N-1} - V_{N-2}}{\Delta x^2}$$

so that

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

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 very easy to program and hard to make mistakes.
- When it does go unstable it is usually obvious.
- It copes well with coefficients that are asset and/or time dependent.
- It is easy to incorporate accurate one-sided differences.

Disadvantage of explicit schemes

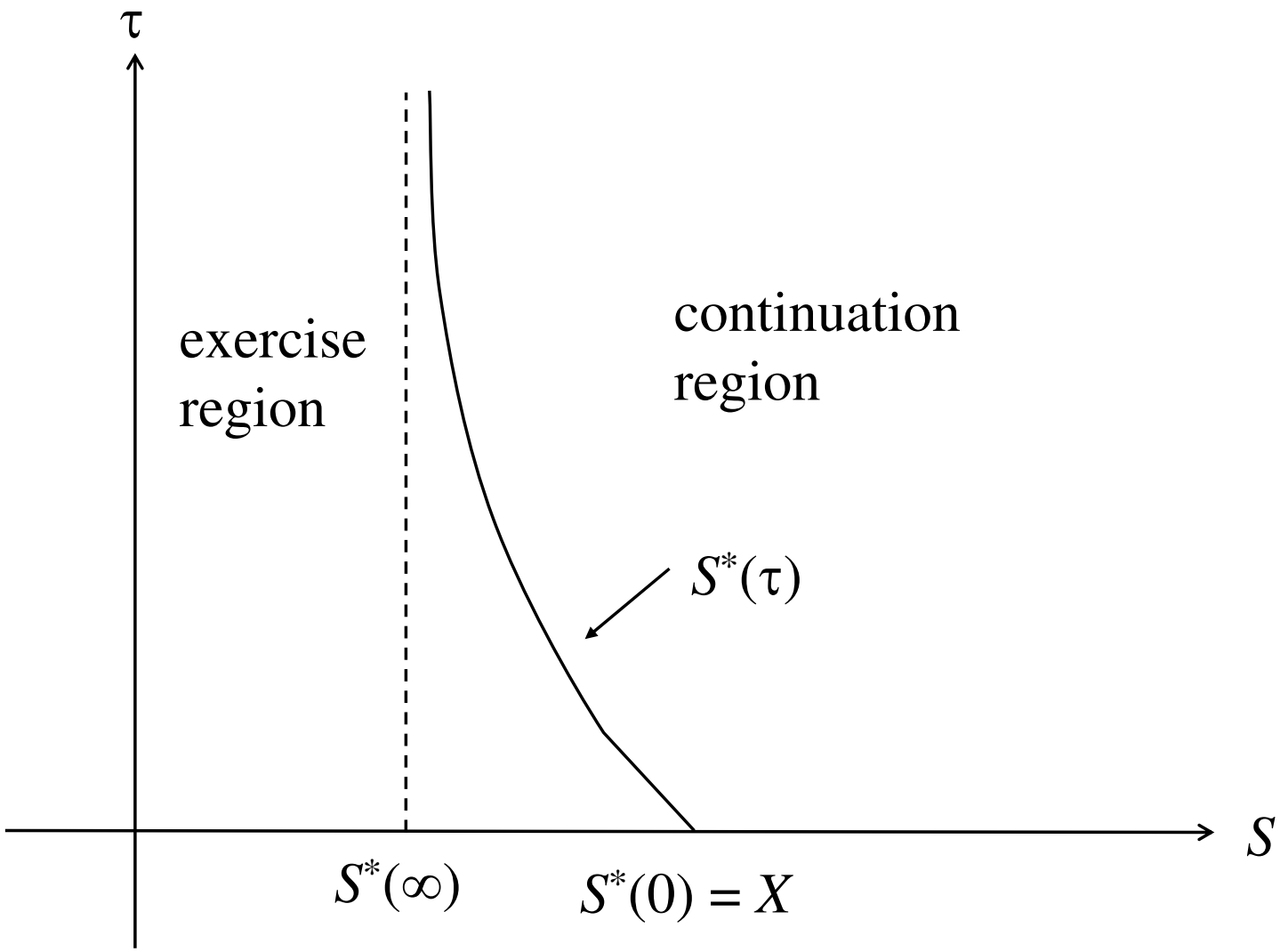
- There are restrictions on the time step due to numerical stability consideration so the method would be less efficient than other schemes.

2.2 Pricing of American options

- The difficulties in the construction of numerical algorithms for solving American style option models arise from the unknown optimal exercise boundary (which has to be obtained as part of the solution).

- 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 or otherwise. If this is the case, the intrinsic value is taken as the option value (signifying the early exercise of the American option).

- Difficulties in implementing the above dynamic programming procedure are encountered when an implicit scheme is employed since option values are obtained implicitly.



Characterization of the continuation region and exercise region of an American put on a non-dividend paying asset. The optimal exercise price $S^*(\tau)$ is a free boundary that separates the continuation and exercise regions.

- The optimal exercise boundary $S^*(\tau)$ separates the continuation region (the American put remains alive) and the exercise region (the American put should be optimally exercised).

- The regret of early exercise becomes lower when the stock price falls to a lower value since the chance that the put expires in-the-money is higher.

- When the stock price falls to the optimal exercise price $S^*(\tau)$ (note the time dependence), the loss in insurance value associated with holding the American option is compensated by the gain in the time value of the strike price received earlier.

For simplicity, we take the strike price to be unity. In the continuation region, the put value $P(S,\tau)$ satisfies the Black-Scholes equation

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

subject to the boundary conditions:

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

and initial condition:

$$P(S,0) = 0 \quad \text{for} \quad S^*(0) < S < \infty, \text{ with } S^*(0) = 1.$$

The difficulty lies in the discretization of the computational domain since the domain of definition of the continuous model has the unknown free boundary $S^*(\tau)$ at its left end.

- The smooth pasting condition

$$\frac{\partial P}{\partial S}(S^*(\tau), \tau) = -1$$

  represents the optimality of early exercise at $S = S^*(\tau)$. This optimality condition provides an additional auxiliary condition for the determination of the free boundary $S^*(\tau)$ (not known aprior).

- The zero terminal payoff condition indicates that the American put (underlying asset is non-dividend paying) is worthless if it is held optimally until expiry.

Front fixing method

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. Recall that $S^*(\tau)$ is not known aprior but it has to be determined as part of the solution.

In terms of the new independent variable y, and observing $\frac{\partial P}{\partial \tau} = \left.\frac{\partial P}{\partial \tau}\right|_y$ $+ \frac{\partial P}{\partial y}\frac{\partial y}{\partial \tau} = \frac{\partial P}{\partial \tau} - \frac{S^{*\prime}(\tau)}{S^*(\tau)}\frac{\partial P}{\partial y}$, the above governing equation becomes

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

Note that

$$\frac{\partial P}{\partial S} = \frac{1}{S}\frac{\partial P}{\partial y} \text{ so that } \frac{\partial P}{\partial S}(S^*(\tau), \tau) = -1 \text{ becomes}$$

$$\frac{\partial P}{\partial y}(0, \tau) = -S^*(\tau).$$

The new set of auxiliary conditions are given by

$$\begin{aligned} P(0,\tau) &= 1 - S^*(\tau), \frac{\partial P}{\partial y}(0,\tau) = -S^*(\tau), P(\infty,\tau) = 0, \\ P(y,0) &= 0 \quad \text{for } 0 < y < \infty. \end{aligned}$$

- The non-linearity in the American put model is revealed by the non-linear term $\frac{S^{*'}(\tau)}{S^*(\tau)}\frac{\partial P}{\partial y}$. If both S^* and P are to be determined at the new time level, then we are required to solve a non-linear algebraic equation involving $S^*(\tau_{n+1})$ and $P(y_j, \tau_{n+1})$.

- We derive a relation between $S^*(\tau)$ and $\frac{\partial^2 P}{\partial y^2}(0^+,\tau)$. Along the boundary $y = 0$, we apply the continuity properties 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

$$\frac{\sigma^2}{2}\frac{\partial^2 P}{\partial y^2}(0^+,\tau) = \frac{\partial}{\partial \tau}[1 - S^*(\tau)] - \left(r - \frac{\sigma^2}{2}\right)[-S^*(\tau)]$$
$$+ \; r[1 - S^*(\tau)] - \frac{S^{*'}(\tau)}{S^*(\tau)}[-S^*(\tau)]$$
$$= r - \frac{\sigma^2}{2}S^*(\tau). \qquad \text{(i)}$$

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 the differential equation would result in a non-linear algebraic system of equations for the determination of P_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 maintain the same order of accuracy as that of the Crank-Nicholson scheme, we adopt a three-level scheme of the form

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

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

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

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

$$D_+ D_- P_j^n = \frac{\frac{1}{\Delta y}\left(P_{j+1}^n - P_j^n\right) - \frac{1}{\Delta y}\left(P_j^n - P_{j-1}^n\right)}{\Delta y} = \frac{P_{j+1}^n - 2P_j^n + P_{j-1}^n}{\Delta y^2}.$$

Remarks

- We choose the discretization of $\frac{\partial P}{\partial y}$ and $\frac{\partial^2 P}{\partial y^2}$ to be the average at the $(n-1)^{\text{th}}$ and $(n+1)^{\text{th}}$ time level due to numerical stability consideration.

- In order to maintain $O(\Delta\tau^2)$ accuracy in a two-level implicit scheme, we are then forced to discretize the non-linear term $\frac{S^{*\prime}(\tau)}{S^*(\tau)}\frac{\partial P}{\partial y}$ as

$$\frac{S^*_{n+1} - S^*_{n-1}}{\Delta\tau\left(\frac{S^*_{n+1}+S^*_n}{2}\right)} D_0\left(\frac{P^{n+1}_j + P^n_j}{2}\right).$$

  This leads to various products of unknowns involving $S^*_{n+1}, P^{n+1}_{j-1}, p^{n+1}_j$ and P^{n+1}_{j+1}, and a system of non-linear algebraic equations would be resulted.

The discretization of the value matching condition, smooth pasting condition and the boundary equation (i) lead to the following system of equations that relate P_{-1}^n, p_0^n, P_1^n, and S_n^*:

$$\begin{aligned} & P_0^n = 1 - S_n^* \\ & \frac{P_1^n - P_{-1}^n}{2\Delta y} = -S_n^* \\ & \frac{\sigma^2}{2}\left[\frac{P_1^n - 2P_0^n + P_{-1}^n}{\Delta y^2}\right] + \frac{\sigma^2}{2} S_n^* - r = 0. \end{aligned} \tag{ii}$$

Here, P_{-1}^n is a fictitious value outside the computational domain. By eliminating P_{-1}^n and P_0^n from the above 3 equations, we obtain

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

where

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

- Once P_1^n is known, we can find S_n^* using (iii) and P_0^n using (ii).

- 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 large. Therefore, we choose M to be sufficiently large such that we set $P_{M+1}^n = 0$ with sufficient accuracy.

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

- We define the following parameters

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

where $\mu = \dfrac{\Delta\tau}{\Delta y^2}$, and the tridiagonal matrix

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

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

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

where $g^n = \dfrac{S^*_{n+1} - S^*_{n-1}}{S^*_n}$. By inverting the matrix $(I+A)$, we obtain

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

where

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

Note that P_0^{n+1} and S^*_{n+1} can be expressed in terms of P_1^{n+1}, where

$$S^*_{n+1} = \frac{\alpha - P_1^{n+1}}{\beta} \quad \text{and} \quad P_0^{n+1} = 1 - S^*_{n+1}.$$

Initialization at the first time level

For the three-level scheme, we need $\boldsymbol{P}^1$ in addition to $\boldsymbol{P}^0$ to initialize the computation. To maintain an overall second order accuracy, we employ the following two-step predictor-corrector technique to obtain $\boldsymbol{P}^1$:

$$\begin{aligned}
\left(I+\frac{A}{2}\right)\widetilde{\boldsymbol{P}} &= \left(I-\frac{A}{2}\right)\boldsymbol{P}^0+\frac{b}{2}\tilde{P}_0\boldsymbol{e}_1+\tilde{g}D_0\boldsymbol{P}^0,\\
\left(I+\frac{A}{2}\right)\boldsymbol{P}^1 &= \left(I-\frac{A}{2}\right)\boldsymbol{P}^0+\frac{b}{2}P_0^1\boldsymbol{e}_1+g^1D_0\left(\frac{\widetilde{\boldsymbol{P}}+\boldsymbol{P}^0}{2}\right),
\end{aligned}$$

where the first equation gives the predictor value $\widetilde{\boldsymbol{P}}$ and the corrector value $\boldsymbol{P}_1$ is finally obtained using the second equation. The predictor-corrector approach avoids the occurrence of product of unknown values of S_1^* and $D_0\left(\frac{\boldsymbol{P}^1+\boldsymbol{P}^0}{2}\right)$. The provisional values and g^1 are given by

$$\begin{aligned}
\tilde{P}_0 &= 1-\tilde{S}_0^*, \quad \tilde{S}_0^*=\frac{\alpha-\tilde{P}_1}{\beta},\\
\tilde{g} &= \frac{\tilde{S}_0^*-S_0^*}{S_0^*} \quad \text{and} \quad g^1=\frac{S_1^*-S_0^*}{\frac{\tilde{S}_0^*+S_0^*}{2}}.
\end{aligned}$$

Front fixing method versus explicit scheme

- *Resolution of the optimal exercise boundary* $S^*(\tau)$

  The explicit scheme determines whether a node lies inside or outside the exercise region by the dynamic programming procedure. The FSG method solves for $S^*(\tau)$ directly as part of the solution procedure.

- *Order of accuracy*

  The adoption of the three-level discretization guarantees second order in temporal accuracy.

- *Ease of implementation*

  The implicit FSG scheme requires the solution of a tridiagonal system of equations at every time step. Also, the three-level discretization requires initialization at the first time level using a separate predictor-corrector scheme.

Projected successive-over-relaxation method

The application of the dynamic programming procedure is not quite straightforward in implicit schemes. This is because the continuation value is not explicitly known from the implicit finite difference formula, so it cannot be used to compare with the intrinsic value directly.

The naive approach of computing V_j^{n+1} from the tridiagonal system of equations derived from the Crank-Nicolson scheme, then followed by comparing V_j^{n+1} with the intrinsic value is NOT acceptable since we do not known in advance whether V_{j-1}^{n+1} assumes the intrinsic value or the corresponding continuation value $\left(\text{same for } V_{j+1}^{n+1}\right)$. 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 equation has no information on whether the option values at neighboring nodes assume the continuation value or exercise value.

Consider an implicit finite difference scheme in 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. The *Gauss-Seidel* iterative procedure produces the k^{th} iterate of V_j by

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

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. Hence, 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.

- To accelerate the rate of convergence of the iteration, we multiply the correction term by a relaxation parameter ω.

The corresponding iterative procedure becomes

$$V_j^{(k)} = V_j^{(k-1)} + \frac{\omega}{a_0}\left(d_j - a_{-1}V_{j-1}^{(k)} - a_0 V_j^{(k-1)} - a_1 V_{j+1}^{(k-1)}\right), \quad 0 < \omega < 2.$$

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 given by

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

The successive iterates on option values contain the information on whether they assume the continuation value or exercise value.

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 solution for V_j.

2.3 Numerical approximation of auxiliary conditions

Sources of errors

- The truncation error, which stems from the difference approximation of the differential operators.

- The numerical approximation of the auxiliary conditions in the option models.

Auxiliary conditions refer to the terminal payoff function plus (possibly) additional boundary conditions due to the embedded path dependent features in the option contract.

Smoothing of discontinuities in the 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.

- Set the payoff value at a node point 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

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

  instead of simply taking the value $V_T(S_j)$.

- 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 $\Delta S/8$.

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

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 usual finite difference calculations for subsequent time levels.

Lookback options

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 the 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, $T_0 < t$. The terminal payoff at time T of the lookback call is given by

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

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

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

1. How to justify the boundary condition at $S = m$, where $\left.\dfrac{\partial c}{\partial m}\right|_{s=m} = 0$?

   When $s = m$, the future updating of the realized minimum value does not require the current realized minimum value m. Hence, the call value is insensitive to the current realized minimum value.

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

   We expect that over the infinitesimal time interval dt. the contribution to the change in call value is given by $\dfrac{\partial c}{\partial m}\, dm$. We observe that

   (i) when $S > m, dm = 0$.

   (ii) when $S = m, \dfrac{\partial c}{\partial m} = 0$.

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

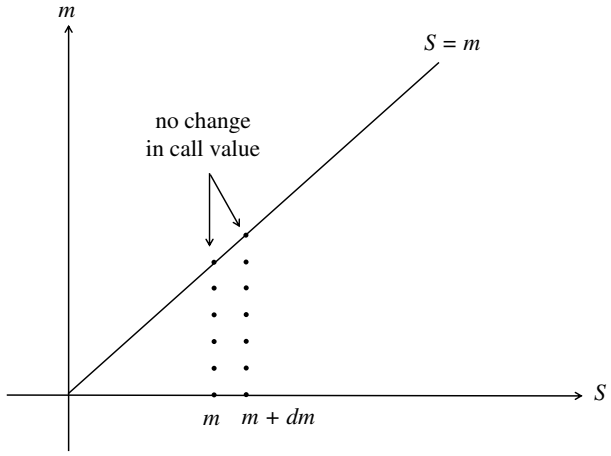
Governing differential equation:

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

with

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

Here, m is a parameter that appears in the auxiliary conditions only



For each m, we solve the differential equation. However, the lookback call values among different values of m, are connected through the boundary condition: $\left.\frac{\partial c}{\partial m}\right|_{S=m} = 0.$

We choose the following set of similarity variables:

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

where $\tau = T - t$, then 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 terminal payoff condition becomes the following initial condition

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

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

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

Using the explicit FTCS scheme, we obtain

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

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

Consider 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 scheme demonstrate $O(\Delta t)$ rate of convergence.

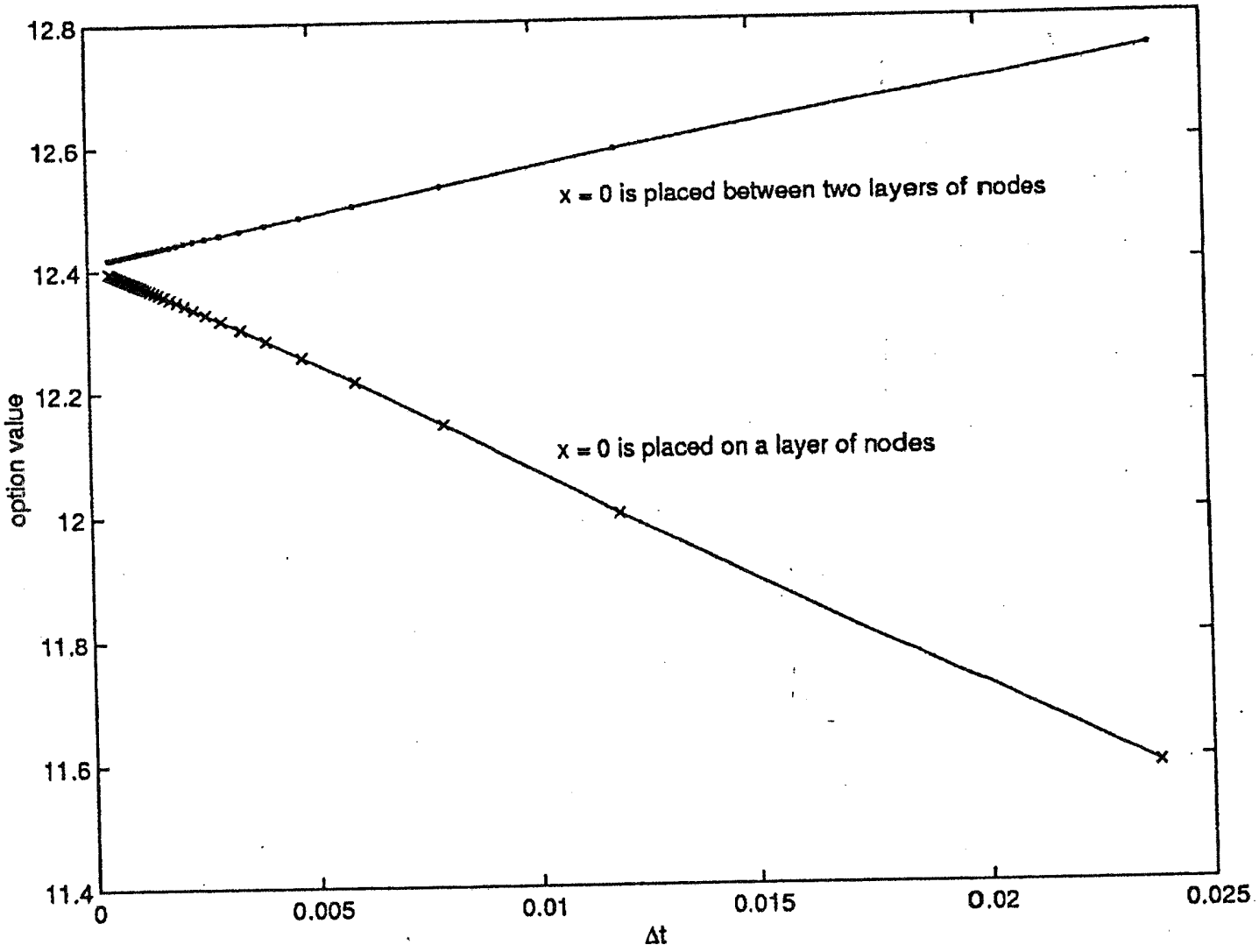
Discretely monitored floating strike lookback call option

The realized minimum is updated only on 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.

- Suppoe the n^{th} time level happens to be a monitoring instant, the boundary condition $\dfrac{\partial V}{\partial x}(0,\tau) = 0, x \leq 0$, is implemented by setting the numerical option values to the left of $x = 0$ to be

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



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

2.4 Properties of numerical solutions

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.

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. The local truncation error at the node point $(j\Delta x, n\Delta\tau)$ of the explict FTCS scheme is given by

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

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

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

- 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.

- 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).

- For the implicit Crank-Nicolson scheme, it is second order time accurate and second order space accurate.

- The highest order of accuracy that can be achieved for a two-level six-point scheme is known to be $O(\Delta\tau^2, \Delta x^4)$.

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 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 $\dfrac{\Delta\tau}{2}$ 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

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

Comparison of finite difference methods

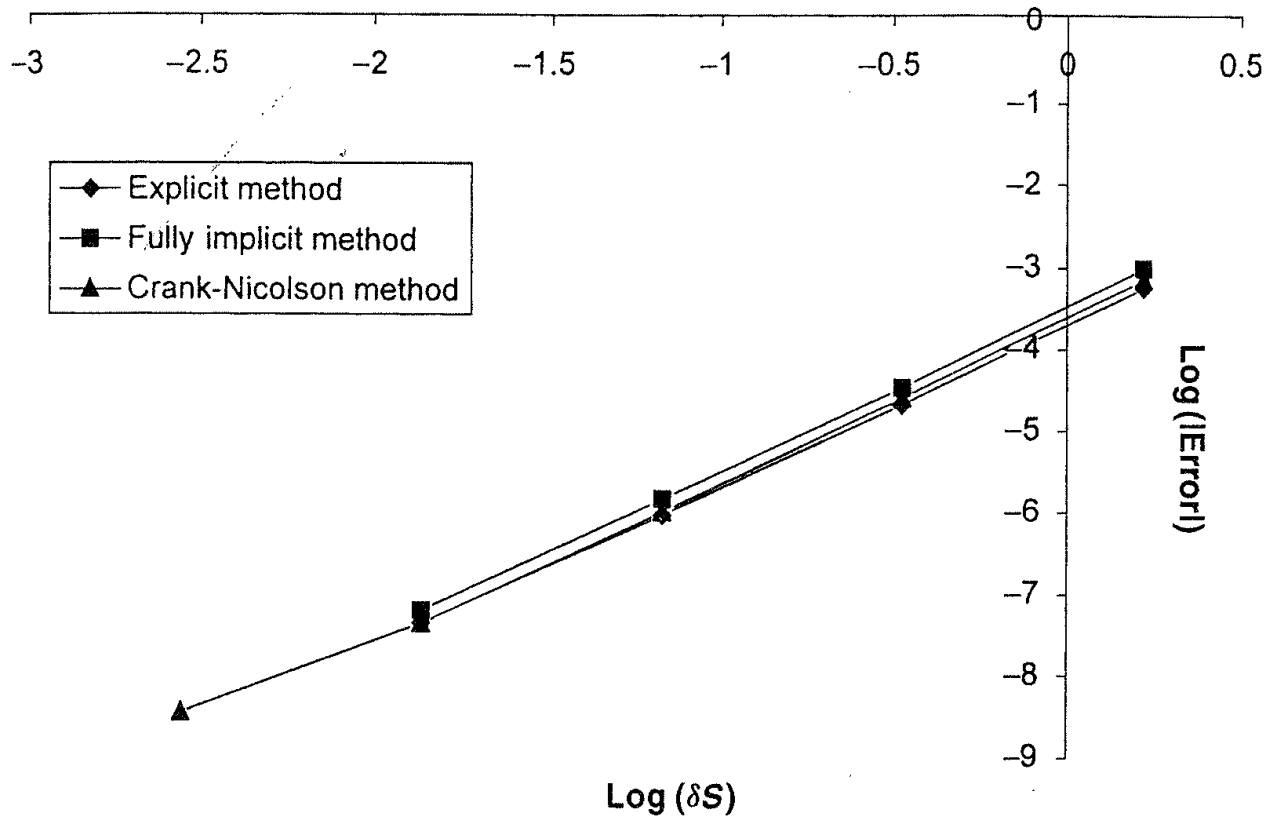
An at-the-money European call option with strike 20, 3 months to expiry, volatility of 20% and interest rate of 5%.

Spatial order of convergence:-

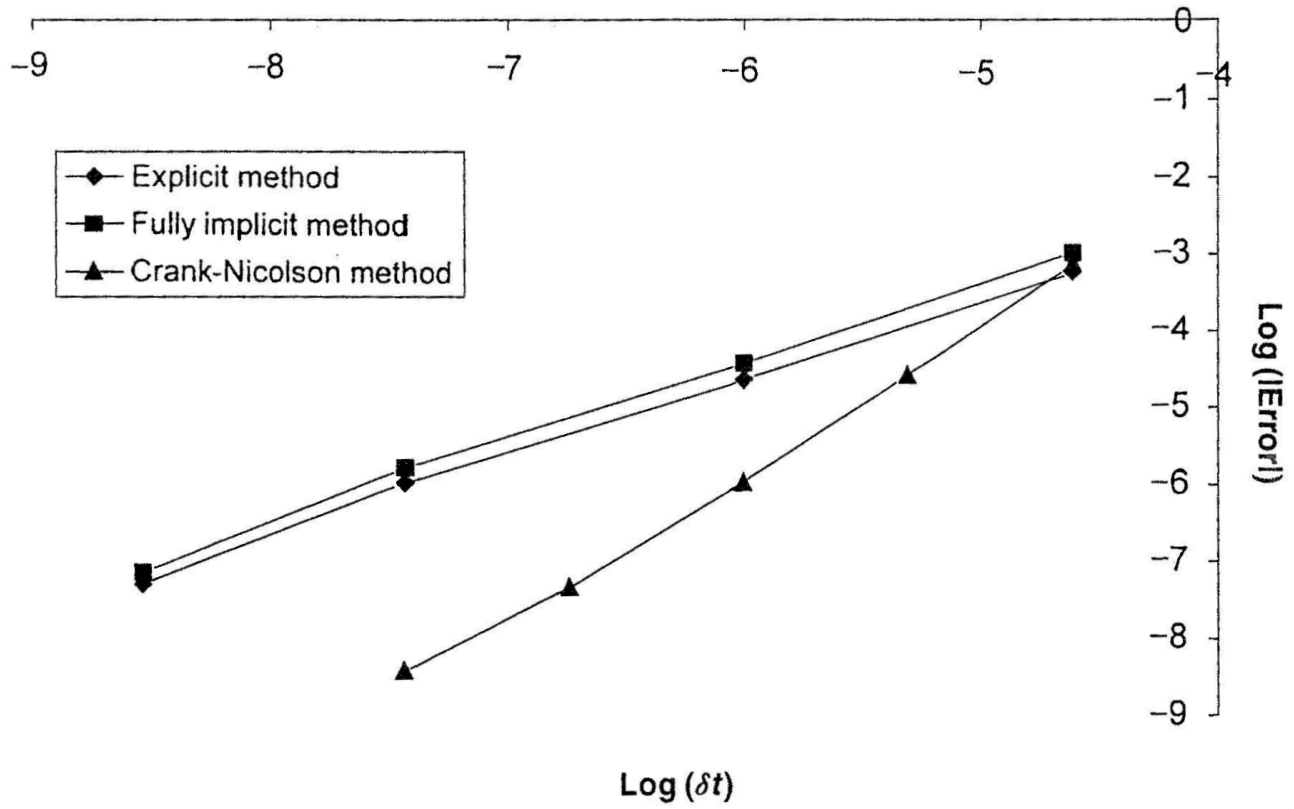
- error in the 3 methods (explicit, fully implicit and Crank–Nicolson) decreases like ΔS^2.

Temporal order of convergence:-

- explicit and fully implicit schemes have an error that decreases like Δt
- Crank–Nicolson's error decreases like Δt^2.



Log(Error) as a function of $\log(\delta S)$ for the three finite-difference schemes.



Log(Error) as a function of log(δt) for the three finite-difference schemes.

Numerical stability

- 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.

- 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 difference scheme.

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 $i = \sqrt{-1}$. 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 von Neumann stability criterion examines the growth of the above Fourier component.

Substituting the Fourier representation 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 $0 \leq k\Delta x \leq \pi$. Henceforth, we write $\beta = k\Delta x$.

Stability of the Cox-Ross-Rubinstein binomial scheme

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

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

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

  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 condition, 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 required conditin coincides with the intuition that probabilities of up jump and down jump cannot be negative.

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.

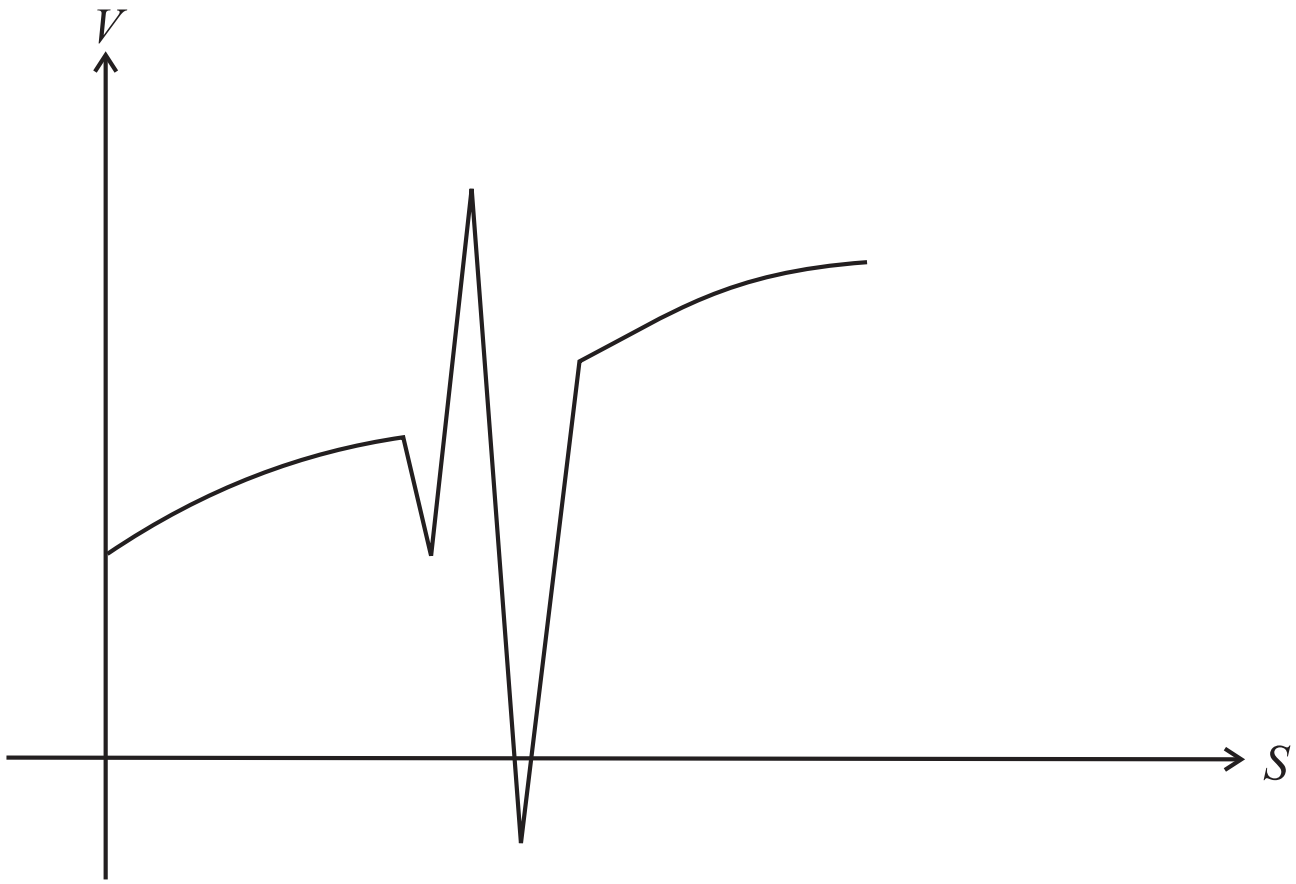
Order of accuracy and stability of Crank-Nicolson scheme

- 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.

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.

Theorem

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 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

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

so

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

since b_{-1}, b_0 and b_1 are non-negative.

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. This alternating sign property can be shown to persist at all later time levels.

In this way, we deduce that

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

We sum over all values of j of the above equation and let $\mathcal{S}^n = \sum_j |V_j^n|$.

As a result, we obtain

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

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

$$\mathcal{S}^n = (1 - 2b_0)^n \mathcal{S}_0 = (1 - 2b_0)^n \varepsilon,$$

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