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
- Projected successive-over-relaxation method
3.2 Numerical approximation of auxiliary conditions

- Initial conditions and terminal payoffs
- Lookback options - Neumann boundary condition
- Barrier type options

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: \[ \frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0. \]

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

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

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

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 $\Delta S$ of $1$ rather than uniform step width in $\ln S$. The increment $\Delta x$ 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 \((j \Delta x, n \Delta \tau), j = 0, 1, \ldots, N + 1, n = 0, 1, 2, \ldots \) with \((N + 1)\Delta x = M_1 + M_2\).

Step width \( \Delta x \) 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 \( \tau \) (time to expiry) in the finite difference calculations.
Finite difference mesh with uniform stepwidth $\Delta x$ and time step $\Delta \tau$. Numerical option values are computed at the node points $(j\Delta x, n\Delta \tau), j = 1, 2, \ldots, N, \; n = 1, 2, \ldots$. 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:

\[
\frac{V^n_{j+1} - V^n_j}{\Delta S}, \quad \frac{V^n_j - V^n_{j-1}}{\Delta S} \quad \text{and} \quad \frac{V^n_{j+1} - V^n_{j-1}}{2\Delta S}.
\]

Approximations to the delta or \(\frac{\partial V}{\partial S}\).
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

\[
\frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} = \left\{ \left[ f(x) + f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 + \frac{f'''(x)}{3!}\Delta x^3 + \frac{f''''(x)}{4!}\Delta x^4 + \cdots \right] - \left[ f(x) - f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 - \frac{f'''(x)}{3!}\Delta x^3 + \frac{f''''(x)}{4!}\Delta x^4 + \cdots \right] \right\} / (2\Delta x)
\]

\[
= f'(x) + \frac{f''''(x)}{6} \Delta x^2 + \cdots
\]

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

\[
f''(x) \approx \frac{f'(x + \frac{\Delta x}{2}) - f'(x - \frac{\Delta x}{2})}{\Delta x}
\]

\[
\approx \frac{[f(x + \Delta x) - f(x)] - [f(x) - f(x - \Delta x)]}{\Delta x^2}
\]

\[
= \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2}.
\]

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 \( f''''(x). \)

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

We obtain the forward difference formula:

\[
\begin{aligned}
f''(x) &\approx \frac{2f(x) - 5f(x + \Delta x) + 4f(x + 2\Delta x) - f(x + 3\Delta x)}{\Delta x^2} + O(\Delta x^2). \\
\end{aligned}
\]

Similarly, the backward difference formula is given by

\[
\begin{aligned}
f''(x) &\approx \frac{2f(x) - 5f(x - \Delta x) + 4f(x - 2\Delta x) - f(x - 3\Delta x)}{\Delta x^2} + O(\Delta x^2). \\
\end{aligned}
\]
Explicit schemes

Let $V^n_j$ 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

\[
\frac{\partial V}{\partial \tau}(j \triangle x, n \triangle \tau) \approx \frac{V_{j+1}^n - V_j^n}{\triangle \tau} \quad \text{(forward difference)}
\]

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

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

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

\[
\frac{W_{j+1}^n - W_j^n}{\triangle \tau} = \frac{\sigma^2 W_{j+1}^n - 2W_j^n + W_{j-1}^n}{2 \triangle x^2} + \left( r - \frac{\sigma^2}{2} \right) \frac{W_{j+1}^n - W_{j-1}^n}{2 \triangle x}.
\]
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:

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

• 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 \( \tau = \Delta \tau \).

• 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

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

The above FTCS scheme corresponds to

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

This resembles the trinomial scheme by observing

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

For example, \( b_1 \) becomes \( \frac{1}{2\lambda^2} + \left( r - \frac{\sigma^2}{2} \right) \frac{\sqrt{\Delta \tau}}{2\lambda \sigma} \), 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 $\Delta x$.

These coefficients 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 $\Delta x$, so that $\Delta x = \ln u$. Similarly, $\ln d = -\Delta x$. Note that $x + \Delta x = \ln S + \ln u = \ln uS$ and $x - \Delta x = \ln dS$. The binomial scheme can be expressed as

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

where $V^{n+1}(x)$, $V^n(x + \Delta x)$ and $V^n(x - \Delta x)$ are analogous to $c$, $c^\Delta u$ and $c^\Delta d$, 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 $\lambda$ 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 $\tau$ is required.
• 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

- 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 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 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,

\[
p_{\text{barrier}}(\ln B, \tau) = R(\tau),
\]

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

We must specify the option value along the two boundaries of the computational domain. 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 $S = 0$, $V^n_0 = 0$.
   
   For large $S$, the call value tends to $S - X e^{-r(T-t)}$.
   
   $$V^n_{N+1} = (N+1) \Delta S - X e^{-rn\Delta \tau}.$$  

2. For a put option, at $S = 0$, $V = X e^{-r(T-t)}$ so that
   
   $V^n_0 = X e^{-rn\Delta \tau}$.

   The put option becomes worthless for large $S$ so that
   
   $V^n_{N+1} = 0$. 
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

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

Almost all common contracts have this property. This is particularly useful because it is independent of the type of a 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:

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

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 \( j = N + 1 \) are not prescribed by any boundary conditions arising from the continuous European vanilla option model.

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

- Rather, we take the option values at \( j = N + 1 \) to remain to be governed by the Black-Scholes equation.

- Recall that we cannot approximate \( \frac{\partial^2 V}{\partial x^2} \big|_{j=N+1} \) 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 one-sided backward difference at the \((N + 1)^{th}\) node along the right boundary:

\[
\frac{\partial V}{\partial x} \bigg|_{j=N+1} \approx \frac{V_{N-1} - 4V_N + 3V_{N+1}}{2\Delta x},
\]

\[
\frac{\partial^2 V}{\partial x^2} \bigg|_{j=N+1} \approx \frac{V_{N+1} - 5V_N + 4V_{N-1} - V_{N-2}}{\Delta x^2}
\]

so that

\[
\frac{V_{N+1}^{n+1} - V_{N+1}^n}{\Delta \tau} = \left( r - \frac{\sigma^2}{2} \right) \frac{V_{N-1}^n - 4V_N^n + 3V_{N+1}^n}{2\Delta x} - rV_{N+1}^n.
\]

\(V_{N+1}^{n+1}\) 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 other schemes.
- The information of the boundary conditions takes 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 $(n + 1)^{th}$ time levels

$$-rV \left( j \Delta x, \left( n + \frac{1}{2} \right) \Delta \tau \right) \approx -\frac{r}{2} \left( V^n_j + V^{n+1}_j \right)$$

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

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

and the temporal derivative by the centered difference

$$\frac{\partial V}{\partial \tau} \left( j \Delta x, \left( n + \frac{1}{2} \right) \Delta \tau \right) \approx \frac{V^{n+1}_j - V^n_j}{\Delta \tau},$$
we then obtain the following two-level implicit finite difference scheme

\[ V_{j}^{n+1} = V_{j}^{n} + \frac{\sigma^2}{2} \frac{\Delta \tau}{\Delta x^2} \left( \frac{V_{j+1}^{n} - 2V_{j}^{n} + V_{j-1}^{n}}{2} \right) \]

\[ + \left( r - \frac{\sigma^2}{2} \right) \frac{\Delta \tau}{2\Delta x} \left( \frac{V_{j+1}^{n} - V_{j-1}^{n}}{2} \right) \]

\[ - r\Delta \tau \left( \frac{V_{j}^{n} + V_{j+1}^{n}}{2} \right), \]

which is commonly known as the \textit{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

\[ a_1 V_{j+1}^{n+1} + a_0 V_{j}^{n+1} + a_{-1} V_{j-1}^{n+1} = b_1 V_{j+1}^{n} + b_0 V_{j}^{n} + b_{-1} V_{j-1}^{n}, \]

\[ j = 1, 2, \ldots, N, \quad n = 0, 1, \ldots. \]
The Crank-Nicolson scheme involves 3 option values at both the $n^{th}$ and $(n + 1)^{th}$ time level.
In order to achieve \( O(\Delta \tau^2) \) accuracy, we approximate \( V, \frac{\partial V}{\partial \tau}, \frac{\partial V}{\partial x}, \frac{\partial^2 V}{\partial x^2} \) at the fictitious intermediate \((n + \frac{1}{2})^{th}\) time level.

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

Relate \( V_j^{n+1}, V_j^{n+1} \) and \( V_j^{n+1} \) (to be computed at the new time level) with \( V_j^n, V_j^n \) and \( V_j^n \) (known values at the old time level).
Suppose the values for \( V^n_j \) are all known along the \( n^{\text{th}} \) time level, the solution for \( V^{n+1}_j \) requires the inversion of a tridiagonal system of equations. The two-level six-point scheme can be represented as

\[
\begin{pmatrix}
a_0 & a_1 & 0 & \cdots & \cdots & 0 \\
a_{-1} & a_0 & a_1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 & a_{-1} & a_0
\end{pmatrix}
\begin{pmatrix}
V^{n+1}_1 \\
V^{n+1}_2 \\
\vdots \\
V^{n+1}_N
\end{pmatrix}
= 
\begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_N
\end{pmatrix},
\]

where

\[
c_1 = b_1 V^n_2 + b_0 V^n_1 + b_{-1} V^n_0 - a_{-1} V^{n+1}_0,
\]

\[
c_N = b_1 V^n_{N+1} + b_0 V^n_N + b_{-1} V^n_{N-1} - a_1 V^{n+1}_{N+1},
\]

\[
c_j = b_1 V^n_{j+1} + b_0 V^n_j + b_{-1} V^n_{j-1}, \quad j = 2, \cdots, N - 1.
\]

For the simple cases, we may assume \( V^{n+1}_0 \) and \( V^{n+1}_{N+1} \) to be known values directly available from the boundary conditions.
**Thomas algorithm – solution of the tridiagonal system**

Consider the solution of the following tridiagonal system of the form

\[-a_j V_{j-1} + b_j V_j - c_j V_{j+1} = d_j, \quad j = 1, 2, \ldots, N,
\]

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 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, \( \ldots \), 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 \( V_{N-3}, V_{N-4}, \ldots, V_2, V_1 \) by successive backward substitution.

Suppose the first \( k \) equations have been reduced to the form

\[
V_j - e_j V_{j+1} = f_j \quad j = 1, 2, \ldots, k.
\]

We use the \( k^{th} \) reduced equation to transform the original \( (k + 1)^{th} \) equation to the same form.
**Gaussian elimination procedure at work**

We use the reduced form of the $k^{\text{th}}$ 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,

$$V_{k+1} - \frac{c_{k+1}}{b_{k+1} - a_{k+1} e_k} V_{k+2} = \frac{d_{k+1} + a_{k+1} f_k}{b_{k+1} - a_{k+1} e_k}.$$
We then deduce the following recurrence relations for $e_j$ and $f_j$:

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

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 $f_j$, $j = 1, 2, \ldots, N$. Once the system is in an upper triangular form, we can solve for $V_N, V_{N-1}, \ldots V_1$, 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:

\[
e_j = \frac{c_j}{b_j - a_j e_{j-1}}, \quad e_0 = 0;
\]

\[
f_j = \frac{d_j + a_j f_{j-1}}{b_j - a_j e_{j-1}}, \quad f_0 = 0.
\]

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 $\Delta x$ 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 condition

- The imposition of the numerical boundary conditions at the old and new time levels will have immediate effect on the numerical option values at the new time level since a linear system of equations are solved at every time step.

- If the option values are not prescribed on the boundaries of the computational domain, then the first and the last equation may contain 3 unknowns. Suppose the first equation contains 3 unknowns $V_1$, $V_2$ and $V_3$, one can eliminate $V_1$ from the first two equations so that the new second equation contains $V_2$ and $V_3$ only. One can then proceed with the Thomas algorithm as usual.
Projected successive-over-relaxation method for pricing American options

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 (same for $V_{j+1}^{n+1}$). In other words, the original tridiagonal system of equations for $V_{1}^{n+1} \ldots V_{N}^{n+1}$ is not the appropriate system of equations for the computation of the continuation values. This is because the system of equation has not incorporated the information on whether the option values at neighboring nodes assume the continuation value or exercise value.
Conversion of implicit evaluation into explicit calculations via an iterative procedure

Consider an implicit finite difference scheme of the form

\[ a_{-1}V_{j-1} + a_0V_j + a_1V_{j+1} = d_j, \quad j = 1, 2, \ldots, N, \]

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:

\[ V_j = \frac{1}{a_0} \left( d_j - a_{-1}V_{j-1} - a_1V_{j+1} \right). \]

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

\[
V_j^{(k)} = \frac{1}{a_0} \left( d_j - a_{-1}V_{j-1}^{(k)} - a_1V_{j+1}^{(k)} \right) \\
= V_j^{(k-1)} + \frac{1}{a_0} \left( d_j - a_{-1}V_{j-1}^{(k)} - a_0V_j^{(k-1)} - a_1V_{j+1}^{(k-1)} \right),
\]

where the last term in the above equation represents the correction made on the \((k - 1)^{th}\) iterate of \( V_j \).
By proceeding sequentially node by node, the dynamic programming procedure of maximizing the option value by either choice of continuation or early exercise can be updated immediately. We start from $j = 1$ and proceed sequentially with increasing value of $j$. When we compute $V_j^{(k)}$ in the $k^{\text{th}}$ iteration, the new $k^{\text{th}}$ iterate $V_j^{(k)}$ is already available while only the old $(k - 1)^{\text{th}}$ iterate $V_j^{(k-1)}$ is known.

**Successive-over-relaxation**

To accelerate the rate of convergence of the iteration, we multiply the correction term by a relaxation parameter $\omega$. The corresponding iterative procedure becomes

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

As a necessary condition for convergence, the relaxation parameter $\omega$ must be chosen between 0 and 2.
We use $0 < \omega < 1$ when the successive iterates bracket the exact solution.

\[ V_j^{(k-1)} \quad \times \quad V_j \quad V_j^{(k)} \]

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

\[
V_j^{(k)} = \max \left( V_j^{(k-1)} + \frac{\omega}{a_0} (d_j - a_{-1}V_{j-1}^{(k)} - a_0V_j^{(k-1)} - a_1V_{j+1}^{(k-1)}), h_j \right)
\]

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.
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 \( V_j \).
3.2 Numerical approximation of auxiliary conditions

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

“Initial” conditions as prescribed by terminal payoffs

At expiry, the option value is just the 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

\[ V_j^0 = \max (j\Delta S - X, 0). \]
Smoothing of discontinuities in the terminal payoff functions

- Most terminal payoff function 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.

- 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. Instead of simply taking the value $V_T(S_j)$, the payoff value at node $S_j$ is given by averaging over 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 $\max(S - X, 0)$ as an example. Suppose we set the strike price $X$ fall exactly on a node point, then $V_T(S_j) = 0$ while the cell-averaged value is $\frac{1}{2} \left( \frac{\Delta S}{2} \right)^2 \sqrt{\Delta S} = \Delta S/8$. 


Averaging the terminal payoff for the vanilla European and American calls can provide a more smooth 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 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

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

The extrapolation procedure works well only if convergence of numerical solution value is not erratic (arising from discontinuity of “initial” condition).
**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.
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 Black-Scholes adjustment.
1. Without smoothing of "initial condition" nor Black-Scholes adjustment

<table>
<thead>
<tr>
<th>$n$</th>
<th>Exact</th>
<th>Binomial</th>
<th>Error ratio</th>
<th>Extrapolated</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>18.47260446</td>
<td>$-0.18552506$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>18.47260446</td>
<td>$-0.05399578$</td>
<td>3.43591784</td>
<td>0.07753350</td>
</tr>
<tr>
<td>40</td>
<td>18.47260446</td>
<td>$-0.00191274$</td>
<td>28.22959146</td>
<td>0.05017031</td>
</tr>
<tr>
<td>80</td>
<td>18.47260446</td>
<td>0.01410255</td>
<td>$-0.13563052$</td>
<td>0.03011785</td>
</tr>
<tr>
<td>160</td>
<td>18.47260446</td>
<td>0.01496727</td>
<td>0.94222596</td>
<td>0.01583199</td>
</tr>
<tr>
<td>320</td>
<td>18.47260446</td>
<td>0.01034461</td>
<td>1.44686708</td>
<td>0.00572194</td>
</tr>
<tr>
<td>640</td>
<td>18.47260446</td>
<td>0.00446058</td>
<td>2.31911889</td>
<td>$-0.00142345$</td>
</tr>
<tr>
<td>1280</td>
<td>18.47260446</td>
<td>$-0.00100635$</td>
<td>$-4.43243516$</td>
<td>$-0.00647328$</td>
</tr>
</tbody>
</table>
2. Smoothing of the terminal call payoff

<table>
<thead>
<tr>
<th>$n$</th>
<th>Exact</th>
<th>Smoothing</th>
<th>Error ratio</th>
<th>Extrapolated</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>18.47260446</td>
<td>0.49641997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>18.47260446</td>
<td>0.25124368</td>
<td>1.97585053</td>
<td>0.00606740</td>
</tr>
<tr>
<td>40</td>
<td>18.47260446</td>
<td>0.12642076</td>
<td>1.98736102</td>
<td>0.00159783</td>
</tr>
<tr>
<td>80</td>
<td>18.47260446</td>
<td>0.06341645</td>
<td>1.99350098</td>
<td>0.00041214</td>
</tr>
<tr>
<td>160</td>
<td>18.47260446</td>
<td>0.03176181</td>
<td>1.99662566</td>
<td>0.00010718</td>
</tr>
<tr>
<td>320</td>
<td>18.47260446</td>
<td>0.01589584</td>
<td>1.99812100</td>
<td>0.00002987</td>
</tr>
<tr>
<td>640</td>
<td>18.47260446</td>
<td>0.00795056</td>
<td>1.99933536</td>
<td>0.00000528</td>
</tr>
<tr>
<td>1280</td>
<td>18.47260446</td>
<td>0.00397364</td>
<td>2.00082367</td>
<td>-0.00000327</td>
</tr>
</tbody>
</table>
3. Black-Scholes adjustment at the first time level

<table>
<thead>
<tr>
<th>$n$</th>
<th>Exact</th>
<th>BS Adjustment</th>
<th>Error ratio</th>
<th>Extrapolated</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>18.47260446</td>
<td>0.08592189</td>
<td>1.95921356</td>
<td>0.00178870</td>
</tr>
<tr>
<td>20</td>
<td>18.47260446</td>
<td>0.04385530</td>
<td>1.98398798</td>
<td>0.00035394</td>
</tr>
<tr>
<td>40</td>
<td>18.47260446</td>
<td>0.02210462</td>
<td>2.00434776</td>
<td>-0.00004795</td>
</tr>
<tr>
<td>80</td>
<td>18.47260446</td>
<td>0.01102834</td>
<td>2.02060394</td>
<td>-0.00011246</td>
</tr>
<tr>
<td>160</td>
<td>18.47260446</td>
<td>0.00545794</td>
<td>2.02281302</td>
<td>-0.00006155</td>
</tr>
<tr>
<td>320</td>
<td>18.47260446</td>
<td>0.00269819</td>
<td>1.98333827</td>
<td>0.00002267</td>
</tr>
<tr>
<td>640</td>
<td>18.47260446</td>
<td>0.00136043</td>
<td>1.92745730</td>
<td>0.00005120</td>
</tr>
<tr>
<td>1280</td>
<td>18.47260446</td>
<td>0.00070582</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lookback options - Neumann boundary conditions

For the floating strike lookback options, by applying an appropriate choice 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 boundary 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

\[
c(S, m, T) = S - m.
\]

Note 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 \( \frac{\partial c}{\partial m} \bigg|_{S=m} = 0 \)?

When \( S = m \), the future updating of the realized minimum value does not require the information of 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 \( m \)?

The contribution to the differential change in call value due to differential change in \( m \) is given by \( \frac{\partial c}{\partial m} \, dm \). Intuitively, we argue that

(i) when \( S > m, \frac{\partial c}{\partial m} = 0 \),
(ii) when \( S = m, \frac{\partial c}{\partial m} = 0 \).

Combining the above results, we conclude that \( \frac{\partial c}{\partial m} \, dm = 0 \).
The following choice of similarity variable is applicable for lookback options whose terminal payoff takes the form: \( c(S, m, 0) = Sf\left(\frac{m}{S}\right) \). One such example is the floating strike lookback call option.

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

\[
\frac{\partial c}{\partial m}\bigg|_{S=m} = 0 \quad \text{and} \quad c(S, m, 0) = S - m.
\]

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 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

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

- The above reduction of dimensionality of the option model can be achieved via the use of $S$ as the numeraire.
Continuously monitored floating strike lookback call option

Using the explicit FTCS scheme, we obtain

\[
\frac{V_{j}^{n+1} - V_{j}^{n}}{\Delta \tau} = \frac{\sigma^2 V_{j+1}^{n} - 2V_{j}^{n} + V_{j-1}^{n}}{2 \Delta x^2} + \left(r - q + \frac{\sigma^2}{2}\right) \frac{V_{j+1}^{n} - V_{j-1}^{n}}{2 \Delta x}
\]

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

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
\[
\left. \frac{\partial V}{\partial x} \right|_{x=0} \approx \frac{V^n_1 - V^n_{-1}}{2\Delta x},
\]
where \( V^n_{-1} \) 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: \( V^n_1 = V^n_{-1} \), we obtain
\[
V^{n+1}_0 = \alpha V^n_1 + (1 - \alpha) V^n_0.
\]
Numerical results obtained from the above 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 ill-treatment of the numerical boundary condition at \( S = m \).
The convergence trend of the numerical 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 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 $x$ is now $(-\infty, \infty)$ instead of $(0, \infty)$.

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

$$V_j^n = V_0^n, \quad j = -1, -2 \ldots$$
Positioning of $x = 0$

Either placing $x = 0$ at $x_0$ or between $x_0$ and $x_1$. Suppose $x = 0$ is placed between $x_0$ and $x_1$, we then have

$$V_{0}^n = V_{-1}^n = V_{-2}^n = \ldots$$

Note that

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

By substituting $V_{1}^n = V_{0}^n$ in the explicit FTCS scheme at $j = 0$ on a monitoring instant, we obtain

$$V_{0}^{n+1} = \frac{\alpha + \mu}{2}V_{1}^n + \left(1 - \frac{\alpha + \mu}{2}\right)V_{0}^n,$$

for both positioning methods.

- Due to the zero-derivative condition applied at $x = 0$, the lookback option value is expected to level off when $x$ is close to $x_0$. The numerical accuracy of the approximation of the auxiliary condition may become less sensitive to the positioning of the nodes relative to $x = 0$ under discrete monitoring.
The figure shows the plots of numerical option value against time step $\Delta t$ for a discretely monitored floating strike lookback call option.
Barrier type options

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

\[ \tau \]

monitoring instant

\[ \chi \]

 Barrier

“x” denotes a “knock-out” node
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 trinomial 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 \( \rho \) is called the killing rate and \( \tau_B^- \) is the occupational time in the down knock-out region defined by

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

To capture this killing rate feature, we define the damping factor over one time step to be

\[
d_j = \begin{cases} 
  e^{-\rho \Delta t} & \text{if } x_j \leq \ln B \\
  1 & \text{otherwise} 
\end{cases}
\]
There is no damping effect on the option value if the stock price stays above the down-barrier $B$. 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 $\alpha = \sigma^2 \frac{\Delta \tau}{\Delta x^2}$.

**Question**

How do we incorporate the damping effect into the implicit Crank-Nicolson scheme?
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 vanishing time steps for different types of the path-dependent options and monitoring features

<table>
<thead>
<tr>
<th></th>
<th>Barrier between two layers of nodes</th>
<th>Barrier on a layer of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Barrier options</strong></td>
<td>continuously monitored</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discretely monitored</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Lookback options</strong></td>
<td>continuously monitored</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>discretely monitored</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Proportional-step options</strong></td>
<td>continuously monitored</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>discretely monitored</td>
<td>✓</td>
</tr>
</tbody>
</table>
3.3 Properties of numerical finite difference solutions

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 explicit FTCS scheme is given by substituting the exact solution into the explicit scheme:

\[
T(j \Delta x, n \Delta \tau) = \frac{V(j \Delta x, (n+1) \Delta \tau) - V(j \Delta x, n \Delta \tau)}{\Delta \tau} - \sigma^2 \frac{V((j+1) \Delta x, n \Delta \tau) - 2V(j \Delta x, n \Delta \tau) + V((j-1) \Delta x, n \Delta \tau)}{2} \\
- \left( r - \frac{\sigma^2}{2} \right) \frac{V((j+1) \Delta x, n \Delta \tau) - V((j-1) \Delta x, n \Delta \tau)}{2 \Delta x} + rV(j \Delta x, n \Delta \tau).
\]
We then expand each term by performing the Taylor expansion at the node point \((j\Delta x, n\Delta \tau)\).

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

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

\[
T(j\Delta x, n\Delta \tau) = \frac{\Delta \tau}{2} \frac{\partial^2 V}{\partial \tau^2}(j\Delta x, n\Delta \tau) - \frac{\sigma^2}{24} \Delta x^2 \frac{\partial^4 V}{\partial x^4}(j\Delta x, n\Delta \tau)
- \left( r - \frac{\sigma^2}{2} \right) \frac{\Delta x^2}{3} \frac{\partial^3 V}{\partial x^3}(j\Delta x, n\Delta \tau) + O(\Delta \tau^2)
+ O(\Delta x^4).
\]

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 \textit{consistent}.

The \textit{order of accuracy} of a scheme is the order in powers of $\Delta x$ 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 $\Delta x^2$, that is, $\Delta x^2 = \lambda^2 \Delta \tau$ for some finite constant $\lambda$, 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 $\Delta x$ 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 $O(\Delta \tau^2, \Delta x^4)$. 
Extrapolation techniques

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

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

where $V^n_j(0)$ 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 $\frac{\Delta \tau}{2}$ successively,

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

Hence, $V^n_j(0)$ can be estimated using $V^n_j(\Delta \tau)$ and $V^n_j(\frac{\Delta \tau}{2})$ via

$$V^n_j(0) \approx \frac{2^m V^n_j(\frac{\Delta \tau}{2}) - V^n_j(\Delta \tau)}{2^m - 1}.$$

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

$$V^n_j \left( \frac{\Delta \tau}{2} \right) + \left[ V^n_j \left( \frac{\Delta \tau}{2} \right) - V^n_j(\Delta \tau) \right].$$
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: \( I_n + 5I_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5} dx = \int_0^1 \frac{x^{n-1} dx}{\frac{x^n}{n}} \bigg|_0^1 = \frac{1}{n}, \)

and \( \int_0^1 \frac{1}{x+5} dx = \ln |x+5| \bigg|_0^1 = \ln \frac{6}{5}, \) 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 \( 5I_{n-1} < \frac{1}{n} < 6I_{n-1}. \) We then have

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

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

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

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

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

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

\[
I_1 - \hat{I}_1 = (-5)(I_0 - \hat{I}_0),
\]

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

\[
I_n - \hat{I}_n = (-5)^n(I_0 - \hat{I}_0).
\]

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

Implementation: \( I_{n-1} = \frac{-I_n}{5} + \frac{1}{5n}, n = 20, 19, \ldots, 1; I_{20} = 0.0087301587 \). We obtain \( I_0 - \hat{I}_0 = \left( \frac{-1}{5} \right)^n (I_n - \hat{I}_n) \) (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

\[ V^n_j = A^n(k) e^{ikj\Delta x}, \]

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, \( e^{ikj\Delta x} = e^{ikx} \bigg|_{x=j\Delta x} \) 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

\[
V^{n+1}_{j+1} = \frac{A^{n+1}(k)}{A^n(k)} e^{ik\Delta x} A^n(k) e^{ikj\Delta x} = \frac{A^{n+1}(k)}{A^n(k)} e^{ik\Delta x} V^n_j, \\
V^{n+1}_{j-1} = \frac{A^{n+1}(k)}{A^n(k)} e^{-ik\Delta x} V^n_j, \quad V^n_{j+1} = e^{ik\Delta x} V^n_j, \text{ etc.}
\]
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

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

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

\[
G(\beta) = [p e^{i\beta} + (1 - p) e^{-i\beta}] e^{-r \Delta \tau}
\]

\[ = [p (\cos \beta + i \sin \beta) + (1 - p) (\cos \beta - i \sin \beta)] e^{-r \Delta \tau} \]

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

The von Neumann stability condition requires

\[
|G(\beta)|^2 = [1 + (q^2 - 1) \sin^2 \beta] e^{-2r \Delta \tau} \leq 1, \quad 0 \leq \beta \leq \pi.
\]
When $0 \leq p \leq 1$, we have $|q| \leq 1$ so that $|G(\beta)| \leq 1$ for all $\beta$. 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

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

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 \( \Delta x \). 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

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

$$E^{n+1} \leq b_{-1}E^n + b_0E^n + b_1E^n = E^n$$

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

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

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 $V_0^0 = \epsilon > 0$ and $V_j^0 = 0, j \neq 0$. At the next time level, $V_{-1}^1 = b_1 \epsilon, V_0^1 = b_0 \epsilon$ and $V_1^1 = b_{-1} \epsilon$, 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 $S^n = \Sigma_j |V_j^n|$, then $S^n = (1 - 2b_0)^nS_0 = (1 - 2b_0)^n\epsilon$. 
Summary - Considerations in the construction of finite difference schemes for pricing options

- Derivation of the model formulation – governing 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 (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). Otherwise, 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. This is based 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 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 unconditional 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 (efficient solution of the associated tridiagonal system of equations).