Today’s Topics

- Computational efficiency
- Convergence
  - consistency
  - stability
- Stability for explicit method: restriction on $\Delta \tau$
- Stability of the fully implicit method
- Positive coefficients avoid oscillations
- Rannacher time stepping
Finite Difference for a LVF Model

Form of \( \alpha_i^n, \beta_i^n \) depends on choice of central, forward, backward weighting

Central Weighting (difference)

\[
\alpha_{i,\text{central}}^n = \left[ \frac{(\sigma_i^n)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{r S_i}{S_{i+1} - S_{i-1}} \right]
\]

\[
\beta_{i,\text{central}}^n = \left[ \frac{(\sigma_i^n)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{r S_i}{S_{i+1} - S_{i-1}} \right].
\]

(1)

\[
\sigma_i^n = \sigma(S_i, \tau^n)
\]
Finite Difference for a LVF Model

Forward difference

\[
\alpha_{i,\text{forward}}^n = \frac{(\sigma_i^n)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} \\
\beta_{i,\text{forward}}^n = \left[ \frac{(\sigma_i^n)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{rS_i}{S_{i+1} - S_i} \right].
\] (2)

Backward difference

\[
\alpha_{i,\text{backward}}^n = \left[ \frac{(\sigma_i^n)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{rS_i}{S_i - S_{i-1}} \right] \\
\beta_{i,\text{backward}}^n = \frac{(\sigma_i^n)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})}
\] (3)

Note: discrete equations have same form, regardless of choice of central, forward, backward, only definition of \(\alpha, \beta\) changes
Matrix Form of Equations for a LVF Model

Define the vectors

$$V^{n+1} = \begin{bmatrix} V_0^{n+1} \\ V_1^{n+1} \\ \vdots \\ V_m^{n+1} \end{bmatrix}; \quad V^n = \begin{bmatrix} V_0^n \\ V_1^n \\ \vdots \\ V_m^n \end{bmatrix}, \quad (4)$$

Let $M^n$ be the tridiagonal matrix with entries

$$[M^n V^n]_i = -\Delta \tau \alpha^n_i V_{i-1}^n + \Delta \tau (\alpha^n_i + \beta^n_i + r)V_i^n - \Delta \tau \beta^n_i V_{i+1}^n.$$

Fully implicit timestepping for a European option can be written as

$$[I + M^{n+1}]V^{n+1} = V^n.$$
Crank-Nicolson: Matrix Form for a LVF Model

Let matrix $\hat{M}$ be defined so that (row $i$)

$$
\left[ \hat{M}^n V^n \right]_i = -\frac{\Delta \tau \alpha_i^n}{2} V_{i-1}^n + \frac{\Delta \tau (\alpha_i^n + \beta_i^n + r)}{2} V_i^n - \frac{\Delta \tau \beta_i^n}{2} V_{i+1}^n.
$$

Then, C-N timestepping (European option) can be written as

$$
\left[ I + \hat{M}^{n+1} \right] V^{n+1} = \left[ I - \hat{M}^n \right] V^n
$$

What is $\sigma(S,t) = \sigma(S)$? How to achieve efficiency?
Solving Linear System

To solve numerically

\[ Ax = b \]

we can compute a lower triangular matrix \( L \) and upper triangular matrix \( U \) such that

\[ A = LU \]

In general, row/column permutation is needed to ensure stability. Permutation is not necessary when \( A = (a_{ij}) \) is diagonally dominant (which is the case for implicit/Crank-Nicolson method), i.e.,

\[ |a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i \]

Assume that \( A \) is a tridiagonal matrix (with diagonal dominance).
(Actually the off-diagonal elements are all nonpositive here).

\[
\begin{bmatrix}
  d_1 & f_2 & 0 & 0 & 0 \\
  e_1 & d_2 & f_3 & 0 & 0 \\
  0 & e_2 & d_3 & f_4 & 0 \\
  0 & 0 & e_3 & d_4 & f_5 \\
  0 & 0 & 0 & e_4 & d_5 \\
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
\end{bmatrix}
=
\begin{bmatrix}
  b_1 \\
  b_2 \\
  b_3 \\
  b_4 \\
  b_5 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  d_1 & f_1 & 0 & 0 & 0 \\
  e_2 & d_2 & f_2 & 0 & 0 \\
  0 & e_3 & d_3 & f_3 & 0 \\
  0 & 0 & e_4 & d_4 & f_4 \\
  0 & 0 & 0 & e_5 & d_5 \\
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
\end{bmatrix}
\]

Matlab functions \texttt{spdiags}, \texttt{lu}, \texttt{\textbackslash} the cost is roughly $8(m + 1)$ for such tridiagonal systems.

\textbf{Note.} It is important to input the coefficient matrix as a sparse matrix
Stability and Convergence

Error occurs due to

- Floating point arithmetic errors
- Errors due to approximating derivatives by finite differences
- Propagation error: error made at $\tau^n$ has impact on accuracy of the computed value at $\tau^{n+1}$. 
A FD method is **convergent** if

\[
\lim_{\Delta S, \Delta \tau \to 0} (V^i_n - V(S_i, \tau^n)) = 0.
\]

for all \(0 \leq i \leq N, 0 \leq j \leq m\)

Convergence of a FD method depends on

- local truncation error converges to zero as \(\Delta \tau, \Delta S \to 0\) and
- stability of the FD method.
Consider Black-Scholes equation,

$$V_{\tau} - \mathcal{L}V = 0$$

Consider a smooth function $\psi$ such that $\psi$ has bounded derivatives w.r.t. $(S, \tau)$. 
DEFINITION 0.1 (Truncation Error) The Truncation Error (T.E.) of a discretization of the B.S. equation is 

\[ T.E. = \left[ \{(\psi_\tau)_i^n - (\mathcal{L}\psi)_i^n\}_{\text{discrete}} \right] - \left[ \{\psi_\tau_0^n - (\mathcal{L}\psi)_i^n\} \right]_i \]
Let

\[ \Delta S = \max_i (S_{i+1} - S_i) \]

\[ \Delta \tau = \max_n (\tau^{n+1} - \tau^n) \]

**DEFINITION 0.2 (Consistency)**  A discretization of the B.S. equation is consistent if

\[ \text{Truncation Error} \to 0 \]

As \( \Delta S, \Delta \tau \to 0 \)
Consistency Example

Consider heat equation:

\[ U_\tau = U_{SS} \quad (5) \]

which we discretize by

\[ \frac{U_{i}^{n+1} - U_{i}^{n}}{\Delta \tau} = \frac{U_{i-1}^{n} - 2U_{i}^{n} + U_{i}^{n}}{(\Delta S)^2} \quad (6) \]

Let \( \psi(S, \tau) \) be a smooth test function, then

\[ \frac{\psi_{i}^{n+1} - \psi_{i}^{n}}{\Delta \tau} = [\psi_{\tau}]_{i}^{n} + O(\Delta \tau) \]

\[ \frac{\psi_{i-1}^{n} - 2\psi_{i}^{n} + \psi_{i-1}^{n}}{(\Delta S)^2} = [\psi_{SS}]_{i}^{n} + O((\Delta S)^2) \quad (7) \]

\[ \Rightarrow \frac{\psi_{i}^{n+1} - \psi_{i}^{n}}{\Delta \tau} - \left( \frac{\psi_{i-1}^{n} - 2\psi_{i}^{n} + \psi_{i-1}^{n}}{(\Delta S)^2} \right) - [\psi_{\tau} - \psi_{SS}]_{i}^{n} = O(\Delta \tau) + O((\Delta S)^2) \quad (8) \]
A necessary condition for convergence is that the truncation error goes to zero as $\Delta \tau, \Delta S \to 0$. Such methods are said to be consistent.

Convergence rate:

- $O(\Delta \tau, (\Delta S)^2)$ for explicit/implicit method
- $O((\Delta \tau)^2, (\Delta S)^2)$ for Crank-Nicolson method

Heat equation example: (6) is a consistent discretization of (5).
A FD method is stable if the amplification factor in error propagation is no larger than one.

Stability is a necessary and sufficient condition for convergence if the local truncation error goes to zero as $\Delta \tau, \Delta S \to 0$. 
Stability requirement can impose a condition on the choice of the stepsize.

For a heat equation, the explicit method is \textbf{stable} if

$$0 < \frac{\Delta \tau}{(\Delta x)^2} \leq \frac{1}{2}$$

\begin{tabular}{|c|c|c|c|}
\hline
S & $\frac{\Delta \tau}{(\Delta x)^2} = 0.5$ & $\frac{\Delta \tau}{(\Delta x)^2} = 0.52$ & exact \\
\hline
10 & 0.4419 & 625.0347 & 0.4420 \\
11 & 0.1607 & -457.3122 & 0.1606 \\
\hline
\end{tabular}

WDH: puts with $K = 10$, $r = 5\%$, $\sigma = 20\%$, $T = 1/2$. 
Consider the explicit method for the Black-Scholes equation

\[ V_{i}^{n+1} = V_{i}^{n} (1 - (\alpha_{i} + \beta_{i} + r)\Delta \tau) + \alpha_{i} \Delta \tau V_{i-1}^{n} + \beta_{i} \Delta \tau V_{i+1}^{n} \]

- **Basic test for stability**: require that the discrete value of the option is finite for fixed \( T, S_{\text{max}} \) as \( \Delta \tau, \Delta S \) goes to zero (Thus the number of steps goes to +\( \infty \)).
Requirement for stability is simply that for a finite computational domain \((S \in [0, S_{\text{max}}], \tau \in [0, T])\), \(V_i^n\) remains bounded as \(n \to \infty\).

At the boundary \(S_{\text{max}}\)

\[ V_{m+1}^n = V_m^n \]
We will assume that upstream or central weighting is selected so that

\[ \alpha_i \geq 0 \]

\[ \beta_i \geq 0. \] \hspace{1cm} (9)

From the FD equation and nonnegativity of \( \alpha_i \) and \( \beta_i \), for explicit method, we obtain

\[ |V_{i}^{n+1}| \leq |V_i^n (1 - (\alpha_i + \beta_i + r)\Delta \tau)| + |V_{i-1}^n|\alpha_i \Delta \tau + |V_{i+1}^n|\beta_i \Delta \tau \]
Now assume that

$$1 - (\alpha_i + \beta_i + r)\Delta\tau \geq 0 \quad ; \quad \forall i,$$

(10)

Then

$$|V_{i}^{n+1}| \leq |V_{i}^{n}| (1 - (\alpha_i + \beta_i + r)\Delta\tau) + |V_{i-1}^{n}|\alpha_i\Delta\tau + |V_{i+1}^{n}|\beta_i\Delta\tau.$$

Let

$$\|V^{n}\|_{\infty} = \max_{i} |V_{i}^{n}|.$$

Then we have

$$|V_{i}^{n+1}| \leq \|V^{n}\|_{\infty} (1 - r\Delta\tau)$$

$$\leq \|V^{n}\|_{\infty}, \quad \text{for any } i$$
Note that the above clearly holds for $S = 0$. Hence

$$\|V^{n+1}\|_\infty \leq \|V^n\|_\infty.$$  \hfill (11)

Now, assuming that $\|V^0\|$ is bounded equation (11) says that the discrete solution is bounded as $n \to \infty$.

In other words, the discretization is *stable*.
The key stability condition is equation (10), which can be rewritten as

$$\Delta \tau \leq \frac{1}{\alpha_i + \beta_i + r} ; \quad \forall i,$$

or

$$\Delta \tau \leq \min_i \left( \frac{1}{\alpha_i + \beta_i + r} \right).$$

Note that the above argument establishes that condition (12) is sufficient for stability. It can also be shown that condition (12) is a necessary condition.
Stability for the Explicit Method

For the explicit method, stability restriction on the stepsize limits computational efficiency.

- The time stepsize limitation can be severe if refinement discretization is used near discontinuity.
- A binomial or tri-nominal method is algebraically identical to an explicit finite difference scheme, with a maximum time stepsize for stability.
Stability of Implicit Methods

Implicit

\[ V_{i}^{n+1} = V_{i}^{n} + \Delta \tau \left[ \alpha_{i}(V_{i-1}^{n+1} - V_{i}^{n+1}) + \beta_{i}(V_{i+1}^{n+1} - V_{i}^{n+1}) - rV_{i}^{n+1} \right] \]

where \( \alpha_{i}, \beta_{i} \) are selected from either central or upstream weighting to ensure \( \alpha_{i} \geq 0; \beta_{i} \geq 0 \).

We can similarly analyze the stability of the fully implicit method.
We can write equation the equation equivalently as

\[ V_{i}^{n+1} [1 + \Delta \tau (r + \alpha_i + \beta_i)] = V_{i}^{n} + \Delta \tau \alpha_i V_{i-1}^{n+1} + \Delta \tau \beta_i V_{i+1}^{n+1}. \]

Noting that \( \alpha_i, \beta_i, r \) are all nonnegative, it follows that

\[ |V_{i}^{n+1}| [1 + \Delta \tau (r + \alpha_i + \beta_i)] \leq |V_{i}^{n}| + |V_{i-1}^{n+1}| \Delta \tau \alpha_i + |V_{i+1}^{n+1}| \Delta \tau \beta_i. \]

Thus

\[ |V_{i}^{n+1}| [1 + \Delta \tau (r + \alpha_i + \beta_i)] \leq \|V^{n}\|_{\infty} + \Delta \tau (\alpha_i + \beta_i) \|V^{n+1}\|_{\infty}. \]
Since this is true for all $i$, letting $|V^{n+1}_i| = \|V^{n+1}\|_\infty$, we have

$$\|V^{n+1}\|_\infty [1 + \Delta \tau (r + \alpha_i^* + \beta_i^*)]$$

$$\leq \|V^n\|_\infty + \Delta \tau (\alpha_i^* + \beta_i^*) \|V^{n+1}\|_\infty$$

$$\Rightarrow \|V^{n+1}\|_\infty \leq \frac{\|V^n\|_\infty}{1 + r \Delta \tau} \leq \|V^n\|_\infty.$$ 

Therefore a fully implicit method is \textit{unconditionally stable}, i.e., there is no restriction on timestep size.
We can write fully implicit or Crank-Nicolson timestepping in one equation

\[ [I + (1 - \theta)M]V^{n+1} = [I - \theta M]V^n \]  \hspace{1cm} (13)

- \( \theta = 1/2 \) for Crank-Nicolson
- \( \theta = 0 \) for fully implicit.
It can be shown that Crank-Nicolson timestepping satisfies the necessary and sufficient conditions for *unconditional stability*.

See course notes §19.4.
Fully Implicit Discretization

- Fully implicit method is *unconditionally stable*
  - Does not require timesteps tied to grid size
  - No problem with local refinement

Note that a fully implicit method can be written:

\[
V_{i}^{n+1} = \frac{V_{i}^{n} + \Delta \tau \alpha_{i} V_{i-1}^{n+1} + \Delta \tau \beta_{i} V_{i+1}^{n+1}}{1 + \Delta \tau (r + \alpha_{i} + \beta_{i})}
\]
Coefficients on rhs of equation are

- Nonnegative \((\alpha_i, \beta_i \geq 0)\)
- Sum to less than one

Why is this significant?
Significance of Positive Coefficients

This means that (see notes):

- $V_{i}^{n+1}$ bounded by
  - Max/Min neighboring values (new timestep),
  - Value at node $i$, old timestep

- This means that spurious oscillations
  - Cannot occur!
Note that when

\[ 1 - (\alpha_i + \beta_i + r)\Delta\tau \geq 0 \]

(which is also the stability condition for an explicit method), an explicit method

\[ V_{i+1}^{n+1} = V_i^n (1 - (\alpha_i + \beta_i + r)\Delta\tau) + \alpha_i\Delta\tau V_{i-1}^n + \beta_i\Delta\tau V_{i+1}^n \]

is also a positive coefficient discretization
Crank-Nicolson Timestepping

- C-N is unconditionally stable

Note that the C-N discretization can be written

\[ V_{i}^{n+1} \left( 1 + \frac{\Delta \tau}{2} (\alpha_i + \beta_i + r) \right) = (V_{i-1}^{n} + V_{i-1}^{n+1}) \frac{\alpha_i \Delta \tau}{2} \]

\[ + (V_{i+1}^{n} + V_{i+1}^{n+1}) \frac{\beta_i \Delta \tau}{2} \]

\[ + V_{i}^{n} (1 - \frac{\Delta \tau}{2} (\alpha_i + \beta_i + r)) \]
C-N will not be a positive coefficient discretization unless

$$\Delta \tau \leq \frac{2}{\alpha_i + \beta_i + r} \quad \forall i$$

This is twice the size of the maximum stable explicit timestep size.
C-N: stable, but not positive

- In many cases
  - Non-positivity not serious

- But in some cases (barriers, digital payoffs)
  - C-N timestep > twice explicit timestep size
  - Oscillations may be introduced into solution

- Oscillations may be small in value (see Figure 15.1 and Table 15.1 in the course notes)
  - Magnified when computing delta ($V_S$) and gamma ($V_{SS}$)
Note:

- Positive coefficient $\rightarrow$ oscillations \textit{guaranteed} not to occur.
- C-N discretization ($>\,$ twice explicit timestep) $\rightarrow$ oscillations \textit{may} occur (we don’t know)

But:

- We would like to use C-N as much as possible. ($O((\Delta \tau)^2)$)
- Next we discuss methods for suppressing C-N oscillations
Oscillations in C-N Timestepping

For C-N methods, if timestep \( > \) twice explicit timestep size, C-N not a positive coefficient method. Normally, does not cause difficulties.

But, possible problems for:

- Digital payoffs
- Barriers
- Delta \((V_S)\), gamma \((V_{SS})\) at the strike (usual payoff)

In these cases, we can observe:

- Slow convergence (not at the second order rate)
- Obvious oscillations
Is there a fundamental problem?

Main result from finite element analysis:

- The discrete FE solution converges at a second order rate, at any finite time $\tau_0 > 0$, for any payoff which has a finite number of discontinuities
  - As long as a few modifications to the algorithm are made
Rannacher, Numerische Mathematik, 1984

- The payoff conditions must be smoothed appropriately
  - The payoff should be approximated by piecewise linear basis functions

- Suppose we have a usual put/call payoff
  - Continuous, but derivative discontinuous

- If there is a node at the strike, *no smoothing is required*, since we have a piecewise linear representation of the payoff
For a discontinuous payoff (e.g. digital)

- Smoothing required
- If equally spaced grid near strike, simple method
  - Replace undefined value by one-half limit from right and left
Smoothing the payoff: digital option

C-N timestepping: Rannacher (1984)
• Payoff smoothed (if required)
• After each rough initial state, we take *two* fully implicit timesteps, and then use C-N thereafter

$\rightarrow$ 2nd order convergence obtained, even for nonsmooth payoff, (quite a remarkable result)
Intuitively:

- Initial smoothing
- Two steps of fully implicit
  \[\rightarrow\] Which is guaranteed \textit{not} to produce oscillations
- Smooths solution sufficiently so that typically C-N does not cause any problems.