1. **Explicit Method.** Consider **backward** FD in time at (S_i, t_j)

(1) $$V_i^{j-1} = \bar{e}_i^j V_{i-1}^j + \bar{d}_i^j V_i^j + \bar{f}_i^j V_{i+1}^j, \;\; i = 1, 2, \cdots, N-1$$

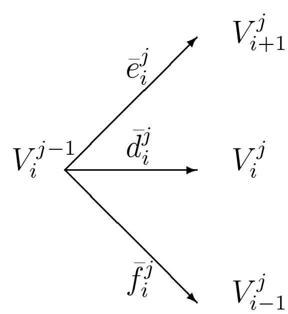
where

$$\begin{aligned}
\bar{e}_i^j &= \frac{1}{2}(\sigma_i^j)^2 S_i^2 \frac{\delta t}{\delta S^2} - \frac{1}{2}(r-q)S_i \frac{\delta t}{\delta S} \\
\bar{d}_i^j &= 1 - (\sigma_i^j)^2 S_i^2 \frac{\delta t}{\delta S^2} - r\delta t \\
\bar{f}_i^j &= \frac{1}{2}(\sigma_i^j)^2 S_i^2 \frac{\delta t}{\delta S^2} + \frac{1}{2}(r-q)S_i \frac{\delta t}{\delta S}
\end{aligned}$$

The explicit methods proceed in the following fashion:

$$\boxed{\begin{array}{l}
V_i^M = \text{payoff}(S_i), \;\; i = 0, \cdots, N \\
\text{for } \; j = M : -1 : 1 \\
\quad\quad \text{compute } V_0^{j-1} \text{ and } V_N^{j-1} \\
\quad\quad \text{compute } \{V_i^{j-1}\}_{i=1}^{N-1} \text{ from } \{V_i^j\}_{i=0}^N \\
\text{end}
\end{array}}$$

Notice that the value V_i^{j-1} depends only on three values $V_{i-1}^j, V_i^j, V_{i+1}^j$. Graphically,



$\Rightarrow$ connection to a trinomial method if $\bar{e}_i^j, \bar{d}_i^j, \bar{f}_i^j > 0$

From the explicit method, we observe that the V_N^M does not influence the values below the grid points $(S_{N-1}, t_{M-1}), (S_{N-2}, t_{M-2}), \cdots, (S_{N-M}, t_0)$. Thus, if N is chosen large enough such that $(N-M)\delta S > S_0$, then the boundary condition does not affect $V(S_0, 0)$.

- **Cost:** O(MN)
- space requirement is $O(MN)$ if $\{V(S_i, t_j)\}$ is needed; $O(N)$ if only the initial price $V(S_0, 0)$ is wanted.

2. **Stability and Convergence**. (See Option Pricing for Detailed Analysis)

The difference between the PDE and the FD equation is called **truncation error**. For explicit method, at each grid point, the truncation error is $O(\delta t, \delta S^2)$.

A FD method is **convergent** if for all $0 \le i \le N, 0 \le j \le M$

$$\lim_{\delta S, \delta t \to 0} (V_i^j - V(S_i, t_j)) = 0.$$

Convergence of a FD method depends on how error at time t_j grows at t_{j-1}; it depends on the stability of the FD method.

A FD method is **stable** if the rounding errors are not magnified at each subsequent iteration.

The explicit method is **stable** if

$$0 < \frac{\delta t}{\delta S^2} \le \alpha$$

where α is a constant independent of $\delta t, \delta S$ but depend on the coefficients of the PDE. For heat equation $\alpha = \frac{1}{2}$; instability occurs when $\alpha > \frac{1}{2}$.

This puts restriction on the time stepsize.

For the explicit method, the **convergence rate** is $O(\max(\delta t, \delta S^2))$.

3. **Implicit Method**.

Assume that $\{V_i^j\}_{i=1}^{N-1}$ has been computed.

At (S_i, t_{j-1}), using **forward FD** for $\frac{\partial V}{\partial t}$, symmetric central FD for $\frac{\partial^2 V}{\partial S^2}$, and central FD for $\frac{\partial V}{\partial S}$, we obtain

$$\frac{V_i^j - V_i^{j-1}}{\delta t} + \frac{1}{2}(\sigma_i^{j-1})^2 S_i^2 \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta S^2} + (r-q)S_i \frac{V_{i+1}^{j-1} - V_{i-1}^{j-1}}{2\delta S} - rV_i^{j-1} = 0$$

Equivalently

$$e_i^{j-1} V_{i-1}^{j-1} + d_i^{j-1} V_i^{j-1} + f_i^{j-1} V_{i+1}^{j-1} = V_i^j, \quad i = 1, \cdots, N-1$$

where

$$\begin{aligned} e_i^{j-1} &= -\tfrac{1}{2}(\sigma_i^{j-1})^2 S_i^2 \tfrac{\delta t}{\delta S^2} + \tfrac{1}{2}(r-q)S_i \tfrac{\delta t}{\delta S} \\ d_i^{j-1} &= 1 + (\sigma_i^{j-1})^2 S_i^2 \tfrac{\delta t}{\delta S^2} + r\delta t \\ f_i^{j-1} &= -\tfrac{1}{2}(\sigma_i^{j-1})^2 S_i^2 \tfrac{\delta t}{\delta S^2} - \tfrac{1}{2}(r-q)S_i \tfrac{\delta t}{\delta S} \end{aligned}$$

The values at the boundary V_0^j and V_N^j are known. Let $x^{j-1} = [V_1^{j-1}, \cdots, V_{N-1}^{j-1}]$. Then

$$A^{j-1} x^{j-1} = b^j,$$

where A^{j-1} is $(N-1) \times (N-1)$ tridiagonal matrix

$$A^{j-1} = \text{diag}(e^{j-1}(2:N), -1) + \text{diag}(d^{j-1}, 0) + \text{diag}(f^{j-1}(1:N-1), 1)),$$

$e^{j-1} = [e_1^{j-1}; \cdots; e_{N-1}^{j-1}]$, and

$$b^j = \begin{bmatrix} V_1^j - e_1^{j-1} V_0^{j-1} \\ V_2^j \\ \vdots \\ V_{N-2}^j \\ V_{N-1}^j - f_{N-1}^{j-1} V_N^{j-1} \end{bmatrix}$$

Computation Cost.

- A^{j-1} is tridiagonal; this structure needs to be exploited for efficiency
- total work is more than the explicit method but still $O(MN)$

Implicit method is stable and its **convergence rate** is $O(\max(\delta t, \delta S^2))$.

4. **Crank-Nicolson Method.**

Can we improve the convergence rate of the implicit and explicit methods?

Average the explicit and implicit methods: for $i = 1, \cdots, N-1$,

$$\frac{1}{2}\left[\frac{V_i^j - V_i^{j-1}}{\delta t} + \frac{1}{2}(\sigma_i^j)^2 S_i^2 \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta S^2} + (r-q)S_i \frac{V_{i+1}^j - V_{i-1}^j}{2\delta S} - rV_i^j\right] +$$
$$\frac{1}{2}\left[\frac{V_i^j - V_i^{j-1}}{\delta t} + \frac{1}{2}(\sigma_i^{j-1})^2 S_i^2 \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta S^2} + (r-q)S_i \frac{V_{i+1}^{j-1} - V_{i-1}^{j-1}}{2\delta S} - rV_i^{j-1}\right] = 0$$

Or,

$$\frac{1}{2} e_i^{j-1} V_{i-1}^{j-1} + \frac{1}{2}(d_i^{j-1} + 1) V_i^{j-1} + \frac{1}{2} f_i^{j-1} V_{i+1}^{j-1} = \frac{1}{2} \bar{e}_{i-1}^j V_{i-1}^j + \frac{1}{2}(1 + \bar{d}_i^j) V_i^j + \frac{1}{2} \bar{f}_i^j V_{i+1}^j,$$

where $e_i^{j-1}, d_i^{j-1}, f_i^{j-1}$ are the same as for the implicit method and, for $i = 1, \cdots N-1$ and $\bar{e}_i^j, \bar{d}_i^j, \bar{f}_i^j$ are the same as in the explicit method.

V_0^{j-1}, V_N^{j-1} are computed from the boundary conditions and a tridiagonal linear system needs to be solved.

Computation Cost.

- coefficient matrix is tridiagonal
- slightly more work than the implicit FD

Crank-Nicolson method is stable and its **convergence rate** is $O(\max(\delta t^2, \delta S^2))$.

In general:

- implicit method is best in stability
- Crank-Nicolson has the best convergence rate
- explicit method is the simplest to implement

5. **Solvning 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 has to be used. This is not necessary when $A = (a_{ij})$ is diagonally dominant (which is the case for implicit/Crank-Nicolson method if δt is sufficiently small), i.e.,

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

Assume that A is a tridiagonal matrix (with diagonal dominance)

$$\begin{bmatrix} d_1 & f_1 & 0 & 0 & 0 \\ e_2 & d_2 & f_2 & 0 & 0 \\ 0 & e_3 & d_3 & f_3 & 0 \\ 0 & 0 & e_4 & d_4 & f_4 \\ 0 & 0 & 0 & e_5 & d_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

$$\begin{bmatrix} d_1 & f_1 & 0 & 0 & 0 \\ e_2 & d_2 & f_2 & 0 & 0 \\ 0 & e_3 & d_3 & f_3 & 0 \\ 0 & 0 & e_4 & d_4 & f_4 \\ 0 & 0 & 0 & e_5 & d_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ l_2 & 1 & 0 & 0 & 0 \\ 0 & l_3 & 1 & 0 & 0 \\ 0 & 0 & l_4 & 1 & 0 \\ 0 & 0 & 0 & l_5 & 1 \end{bmatrix} \begin{bmatrix} u_1 & f_1 & 0 & 0 & 0 \\ 0 & u_2 & f_2 & 0 & 0 \\ 0 & 0 & u_3 & f_3 & 0 \\ 0 & 0 & 0 & u_4 & f_4 \\ 0 & 0 & 0 & 0 & u_5 \end{bmatrix}$$

By comparing the two sides

$$\begin{array}{lll} (1,1): & d_1 = u_1 & \Longrightarrow \quad u_1 = d_1 \\ (2,1): & e_2 = l_2 u_2 & \Longrightarrow \quad l_2 = e_2/u_1 \\ (2,2): & d_2 = l_2 f_1 + u_2 & \Longrightarrow \quad u_2 = d_2 - l_2 f_1 \\ \vdots & \vdots & \vdots \qquad \vdots \end{array}$$

Matlab functions **TriDiLU**, **LBidiSol**, and **UBidiSol** are privided to compute LU factorization and triangular solves for a tridiagonal matrix; the cost is roughly $8N$.