Chapter 7. Finite Difference Methods (v. A.3)

In earlier chapters, we applied the trinomial tree approach to price options, by using the discounted expected payoff formula under the risk-neutral measure.

The price of a call option at the node (i, j) is computed by finding the present value of the expected payoff in the next step (3 nodes):

$$C_{ij} = e^{-r\Delta}\left(P_u \cdot C_{i+1,j+1} + P_m \cdot C_{i+1.j} + P_d \cdot C_{i+1,j-1}\right),$$

The price of a put option is computed as:

$$\pi_{ij} = e^{-r\Delta}(P_u \cdot \pi_{i+1,j+1} + P_m \cdot \pi_{i+1.j} + P_d \cdot \pi_{i+1,j-1})$$

Applying this one-period model to every node and moving backwards in time, at step at a time, we find the option prices in all nodes, including the node that corresponds to time 0.

What if we want to compute option prices for a range of initial stock prices, and not only for S_0? We could slightly modify the trinomial tree method and value options for a range of initial stock prices. In case of a trinomial tree, the number of nodes of the tree is reduced in every step, when moving backwards in time. On the last step, when we get to time 0, only one node is left, which corresponds to S_0 and it is in this node that we compute the price of the option. By slightly modifying the tree approach, we can construct a rectangular grid and move backwards in time to price the option on the grid. In such a case, in every time-step there will be no reduction in number of nodes. Thus, when we get to time 0, not only we computed the price of the option for the initial stock price S_0, but for a range of other initial stock prices. For this to work, we need to be able to estimate the option values on the lower and upper boundaries of a grid. If we could populate the option value cells for the upper and lower boundaries of the grid, then the method would work. This is the basis for solving PDE's numerically to price options.

Consider pricing of a European call option using the above-mentioned method.

Let the first index i be for the time-variable, and the second index j be used for the price-variable.

Assume N is a large number and we take a partition of the stock price range into (2N+1) parts.

$C_{i,j}$ for $j = -N+1, \dots, N-1$ are 2N+1 call option prices at time i.

Assume that $C_{i,N}$ and $C_{i,-N}$ are known for any $i = 1, \dots, M$. $C_{i,N}$ and $C_{i,-N}$ are the call option prices at time i, for a very large stock price (corresponds to N) and for a very small stock price (corresponds to -N).

Then,

$$C_{ij} = e^{-r\Delta}(P_u \cdot C_{i+1,j+1} + P_m \cdot C_{i+1,j} + P_d \cdot C_{i+1,j-1})$$

can be used to price options in node (i,j) for any $i = 0, \dots, M-1$, and $j = -N+1, \dots, N-1$.

Thus, for every i, we'll have $C_{i,-N+1}, \dots, C_{i,N-1}$ computed (using the above formula) and $C_{i,-N}, C_{i,N}$ will be assumed known (we will show later how to estimate these two option prices for extreme values of stock prices).

Therefore, $\{C_{i,j}\}_{j=-N}^{N}$ can be obtained for any $i = 0, \dots, M$.

Starting at time $i = M$ (terminal time) and moving backwards until $i = 0$, we can use the above-mentioned method and compute $C_{0,-N}, \dots, C_{0,N}$ --- the prices of call options at time $i = 0$, with initial stock prices of $S_{-N}, \dots, S_0, \dots, S_N$ respectively.

The remaining question is: how to estimate the option value on the lower and upper boundaries of the grid; or, in other words, option values that correspond to very small or very large values of the underlying security prices at different times (that is $C_{i,-N}$ and $C_{i,N}$)?

Below we propose two methods to estimate the call and put option values for extreme values of the underlying security prices.

For Call Options

(a) The boundary conditions/values can be used as follows:

(1) $C_{M,j} = (S_{M,j} - K)^+$ for $j = -N, \dots, N$ (this is the payoff condition at maturity)

(2) $C_{i,-N} = C_{i,-N+1}$ for $i = 0, \dots, M$ (since for small s, $\frac{\partial c}{\partial s} = 0 \Rightarrow \frac{C_{i,-N} - C_{i,-N+1}}{S_{i,-N} - S_{i,-N+1}} \approx 0$)

(3) $C_{i,N} = C_{i,N-1} + (S_{i,N} - S_{i,N-1})$, for $i = 0, \dots, M$ (since for large s, $\frac{\partial c}{\partial s} = 1 \Rightarrow \frac{C_{i,N} - C_{i,N-1}}{S_{i,N} - S_{i,N-1}} \approx 1$)

(b) An alternative set of conditions

(1) $C_{M,j} = (S_{M,j} - K)^+$ for $j = -N, \dots, N$ (this is the payoff condition at maturity)

(2) $C_{i,-N} = 0$ for $i = 0, \dots, M$ (for small values of S)

(3) $C_{i,N} = S_{i,N} - K \cdot e^{-r(M-i)\cdot\Delta}$ for $i = 0, \dots, N$ (for large values of S)

For Put Options

(a) Boundary conditions:

(1) $\pi_{M,j} = (K - S_{M,j})^+$ for $j = -N, \dots, N$ (this is the payoff condition at maturity)

(2) $\pi_{i,N} = \pi_{i,N-1}$ for $i = 0, \dots, M$ (since for large S, $\frac{\partial \pi}{\partial S} = 0 \Rightarrow \frac{\pi_{i,N} - \pi_{i,N-1}}{S_{i,N} - S_{i,N-1}} \approx 0$)

(3) $\pi_{i,-N} = \pi_{i,-N+1} - (S_{i,-N} - S_{i,-N+1})$ for $i = 0, \dots, M$ (since for small S, $\frac{\partial \pi}{\partial s} = -1 \Rightarrow \pi_{i,-N} = \pi_{i,-N+1} - (S_{i,-N} - S_{i,-N+1})$)

(b) An alternative set of conditions

(1) $\pi_{M,j} = (K - S_{M,j})^+$ for $j = -N, \dots, N$ (this is the payoff condition at maturity)

(2) $\pi_{i,-N} = (K \cdot e^{-r(M-i)\cdot\Delta} - S_{i,-N})^+ = K \cdot e^{-r(M-i)\cdot\Delta}$, for $i = 0, \dots, M$, assuming $S_{i,-N} = 0$.

(3) $\pi_{i,N} = (K \cdot e^{-r(M-i)\cdot\Delta} - S_{i,N})^+ = 0,$ for $i = 0, \dots, M$.

Now, we will construct the rectangular grid, and use the method described above to solve the PDE on the grid. That is, we will use the PDE approach of pricing options, and numerically solve the Black-Scholes PDE, by implementing the approach described above.

The Black-Scholes PDE is given by:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 c}{\partial S^2} + rS\frac{\partial c}{\partial S} - rc = 0$$

We want to solve it by a numerical approximation. First we'll define various methods of approximating the partial derivatives in the above PDE. Then, we will use those methods to numerically solve the PDE with various terminal conditions.

Assume the change in time from i to $i+1$ is Δt, and the change in the stock price from j to $j+1$ is ΔS. Define the following approximations to derivatives at point (i, j):

Forward difference: $\frac{\partial c}{\partial S} \approx \frac{C_{i,j+1} - C_{i,j}}{\Delta S}, \quad \frac{\partial c}{\partial t} \approx \frac{C_{i+1,j} - C_{i,j}}{\Delta t}$

Backward difference: $\frac{\partial c}{\partial S} \approx \frac{C_{i,j} - C_{i,j-1}}{\Delta S}, \quad \frac{\partial c}{\partial t} \approx \frac{C_{i,j} - C_{i-1,j}}{\Delta t}$

Central difference: $\frac{\partial c}{\partial S} \approx \frac{C_{i,j+1} - C_{i,j-1}}{2\Delta S}, \quad \frac{\partial c}{\partial t} \approx \frac{C_{i+1,j} - C_{i-1,j}}{2\Delta t}$

Second-order derivative: $\frac{\partial^2 c}{\partial S^2} \approx \frac{\frac{C_{i,j+1} - C_{i,j}}{\Delta S} - \frac{C_{i,j} - C_{i,j-1}}{\Delta S}}{\Delta S} \approx \frac{C_{i,j+1} - 2C_{i,j} + C_{i,j-1}}{(\Delta S)^2}$

Using these definitions and methods outlined above, we will now describe three methods for solving PDEs numerically.

We will consider two cases of PDEs: one that is based on the log-stock price process ($X = ln(S)$) and another that is just based on the stock price process, S.

If we transform S into $X = ln(S)$, then, the PDE for S will be transformed to the following PDE:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 c}{\partial x^2} + (r - \frac{\sigma^2}{2})\frac{\partial c}{\partial x} - rc = 0$$

The advantage of using this PDE (over the original Black-Scholes PDE) is that the coefficients for the first and second order derivatives with respect to the underlying are now constants. We will use the PDE for X for demonstration of the three numerical methods.

The three methods we will cover in this section are the following:

- Explicit Finite-Difference Method (EFD),
- Implicit Finite-Difference Method (IFD),
- Crank-Nicolson Finite-Difference Method (C-NFD).

These three methods are obtained when we use either the backward, or the forward, or the central difference for the time-derivative, when we discretize the PDE. The table below summarizes the three cases.

Method	$\boldsymbol{\frac{\partial c}{\partial t}}$	$\boldsymbol{\frac{\partial c}{\partial S}}$	$\boldsymbol{\frac{\partial^2 c}{\partial S^2}}$
EFD	Backward	Central	Standard
IFD	Forward	Central	Standard
C-NFD	Central	Central	Standard

To derive the finite difference schemes, we truncate the S-domain (or the X-domain) at a value S_{max}.

When the PDE for S is used, then the computational domain is: $[0, S_{max}] \times [0, T]$, which is discretized by a uniform mesh with steps ΔS and Δt in order to obtain the values of S at nodes (i, j).

When the PDE for X is used, then the computational domain is: $[X_{min}, X_{max}] \times [0, T]$, which is discretized by a uniform mesh with steps ΔX and Δt in order to obtain the values of X at nodes (i, j).

Explicit Finite-Difference Method (EFD)

We will consider the transformed PDE for $X = ln(S)$ first. The Explicit Finite Difference approximation of the PDE $\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 c}{\partial x^2} + (r - \frac{\sigma^2}{2})\frac{\partial c}{\partial x} - rc = 0$ is given by:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta_t} + \frac{1}{2}\sigma^2 \cdot \frac{C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j-1}}{(\Delta x)^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{C_{i+1,j+1} - C_{i+1,j-1}}{2\Delta x} - rC_{i+1,j} = 0$$

which can be re-written as follows:

$$C_{ij} = P_u \cdot C_{i+1,j+1} + P_m \cdot C_{i+1,j} + P_d \cdot C_{i+1,j-1}, \text{ where}$$

$$\begin{cases} P_u = \Delta_t\left(\dfrac{\sigma^2}{2\Delta x^2} + \dfrac{r - \frac{\sigma^2}{2}}{2\Delta x}\right) \\ P_m = 1 - \Delta_t\dfrac{\sigma^2}{\Delta x^2} - r\Delta_t \\ P_d = \Delta_t\left(\dfrac{\sigma^2}{2\Delta x^2} - \dfrac{r - \frac{\sigma^2}{2}}{2\Delta x}\right) \end{cases}$$

for $j = -N + 1, \dots, N - 1$.

Also, we have the following boundary conditions, described earlier for small and large values of S:

$C_{i,-N} = C_{i,-N+1}$ and $C_{i,N} = C_{i,N-1} + \left(S_{i,N} - S_{i,N-1}\right)$, for $i = 0, \dots, M$.

Putting all above together, we will have the following set of 2N+1 equations:

$$\begin{cases} C_{i,N} = C_{i,N-1} + \left(S_{i,N} - S_{i,N-1}\right) \\ C_{i,N-1} = P_u \cdot C_{i+1,N} + P_m \cdot C_{i+1,N-1} + P_d \cdot C_{i+1,N-2} \\ C_{i,N-2} = P_u \cdot C_{i+1,N-1} + P_m \cdot C_{i+1,N-2} + P_d \cdot C_{i+1,N-3} \\ \vdots \\ C_{i,-N+1} = P_u \cdot C_{i+1,-N+2} + P_m \cdot C_{i+1,-N+1} + P_d \cdot C_{i+1,-N} \\ C_{i,-N} = C_{i,-N+1} \end{cases}$$

which can be written in a matrix form as: $F_i = AF_{i+1} + B_i$, where

$$A = \begin{pmatrix} P_u P_m P_d & 0 & \vdots & \vdots & \vdots & 0 \\ P_u P_m P_d & 0 & \vdots & \vdots & \vdots & 0 \\ 0 \; P_u \; P_m P_d & 0 & \vdots & \vdots & \vdots & 0 \\ \vdots \; \vdots \; \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots \; \vdots \; \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 \; 0 \; \vdots & \vdots & 0 P_u P_m P_d \\ 0 \; 0 \; \vdots & \vdots & 0 P_u P_m P_d \end{pmatrix}, \qquad F_i = \begin{pmatrix} C_{i,N} \\ C_{i,N-1} \\ C_{i,N-2} \\ C_{i,N-3} \\ . \\ . \\ C_{i,-N} \end{pmatrix}, B_i = \begin{pmatrix} S_{i,N} - S_{i,N-1} \\ 0 \\ 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

The goal here is to find $\boldsymbol{F}_0$.

The solution of $F_i = AF_{i+1} + B_i$, which is the vector F_i of option values at time i, is already given explicitly as $\boldsymbol{F_i = AF_{i+1} + B_i}$. That is why the method is called the Explicit F.D. method.

<u>Remark:</u> If we take (i, j) instead of $(i+1, j)$ in the last term of the discretized PDE, we'll get

$$C_{ij} = \frac{1}{1 + r\Delta_t} \cdot [P_u \cdot C_{i+1,j+1} + P_m \cdot C_{i+1,j} + P_d \cdot C_{i+1,j-1}]$$

$$\begin{cases} P_u = \Delta_t \left(\dfrac{\sigma^2}{2\Delta x^2} + \dfrac{r - \dfrac{\sigma^2}{2}}{2\Delta x} \right) \\ P_m = 1 - \Delta_t \dfrac{\sigma^2}{\Delta x^2} \\ P_d = \Delta_t \left(\dfrac{\sigma^2}{2\Delta x^2} - \dfrac{r - \dfrac{\sigma^2}{2}}{2\Delta x} \right) \end{cases}$$

Notice that, $\frac{1}{1+r\Delta_t} \approx e^{-r\Delta_t}$, therefore, the latter method is the same as the trinomial method. We'll take $\Delta x \geq \sigma\sqrt{3\Delta_t}$ in this case when implementing the model as it is proven to result in a stable and converging algorithm. Stability and convergence of algorithms will be discussed later.

Now, consider the original Black-Scholes PDE for S. Suppose now we want to solve the PDE:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 c}{\partial S^2} + rS\frac{\partial c}{\partial S} - rc = 0$$

The discretized version of this PDE (using the explicit method) will be given as follows:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta_t} + r \cdot \left(S_{ij}\right) \cdot \frac{C_{i+1,j+1} - C_{i+1,j-1}}{2\Delta S} + \frac{1}{2}\sigma^2 \cdot (S_{ij})^2 \cdot \frac{C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j-1}}{(\Delta S)^2} - rC_{i+1,j} = 0$$

As can be noticed, the two finite differences ($\frac{\partial^2 c}{\partial S^2}$ and $\frac{\partial c}{\partial S}$) contain S −dependent terms. To solve for the derivative security price, we follow the same steps as above: solve for $C_{i,j}$ in terms of $C_{i+1,j+1}$, $C_{i+1,j}$, $C_{i+1,j-1}$.

There are several choices to parameterize S_{ij} and to generate a grid, of which we will mention two:

1. S_{ij} can be $S_0 u^j$ using $u > 1$, and j ranging from $-N$ to N;
2. S_{ij} can be taken as $j(\Delta S)$ where $S_0 = 0$ and S takes on $\{0, \Delta S, 2\Delta S, \ldots, N\Delta S\}$ values.

We will study the case 2 in more details. In that case, if we choose $j\Delta S = S_{ij}$ $\forall i = 0, \ldots, M,$ then we obtain the following set of equations:

$$C_{ij} = P_u^j \cdot C_{i+1,j+1} + P_m^j \cdot C_{i+1,j} + P_d^j \cdot C_{i+1,j-1}$$

$$\begin{cases} P_u^j = & \Delta_t\left(\frac{r\cdot j}{2}+\frac{\sigma^2 j^2}{2}\right) \\ P_m^j = 1-\Delta_t\cdot(\sigma^2 j^2+r) \\ P_d^j = & \Delta_t\left(-\frac{rj}{2}+\frac{\sigma^2 j^2}{2}\right) \end{cases}$$

for $j = 1, \dots, N-1$.

Also, we have that $C_{i,0} = C_{i,1}$ and $C_{i,N} = C_{i,N-1} + \left(S_{i,N} - S_{i,N-1}\right)$, for $i = 0, \dots, M$. Putting all above together, we will have the following set of equations:

$$\begin{cases} C_{i,N} = P_u^{N-1}\cdot C_{i+1,N} + P_m^{N-1}\cdot C_{i+1,N-1} + P_d^{N-1}\cdot C_{i+1,N-2} + \left(S_{i,N} - S_{i,N-1}\right) \\ C_{i,N-1} = P_u^{N-1}\cdot C_{i+1,N} + P_m^{N-1}\cdot C_{i+1,N-1} + P_d^{N-1}\cdot C_{i+1,N-2} \\ C_{i,N-2} = P_u^{N-2}\cdot C_{i+1,N-1} + P_m^{N-2}\cdot C_{i+1,N-2} + P_d^{N-2}\cdot C_{i+1,N-3} \\ \vdots \\ C_{i,1} = P_u^1\cdot C_{i+1,2} + P_m^1\cdot C_{i+1,1} + P_d^1\cdot C_{i+1,0} \\ C_{i,0} = P_u^1\cdot C_{i+1,2} + P_m^1\cdot C_{i+1,1} + P_d^1\cdot C_{i+1,0} \end{cases}$$

which can be written in a matrix form: $F_i = AF_{i+1} + B_i$, where

$$A = \begin{pmatrix} P_u^{N-1} & P_m^{N-1} & P_d^{N-1} & \vdots & \vdots & \vdots & \vdots & 0 \\ P_u^{N-1} & P_m^{N-1} & P_d^{N-1} & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & P_u^{N-2} & P_m^{N-2} & P_d^{N-2} & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & P_u^{N-3} & P_m^{N-3} & P_d^{N-3} & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & 0 & P_u^1 & P_m^1 & P_d^1 \\ 0 & 0 & \vdots & \vdots & 0 & P_u^1 & P_m^1 & P_d^1 \end{pmatrix}, F_i = \begin{pmatrix} C_{i,N} \\ C_{i,N-1} \\ C_{i,N-2} \\ . \\ . \\ C_{i,0} \end{pmatrix}, B_i = \begin{pmatrix} S_{i,N} - S_{i,N-1} \\ 0 \\ 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

The goal here is to find $\boldsymbol{F}_0$.

The above expression is the explicit solution for C_{ij}, that's why the name - Explicit Finite Difference Method.

Standard results from Linear Algebra imply that the above system $F_i = AF_{i+1} + B_i$ is stable, if and only if, $\parallel A \parallel_\infty \leq 1$.

Heuristically, one may say that when the infinity-norm of the matrix A is less than 1, then F_i converges. There are values of Δ_t and σ for which $\parallel A \parallel_\infty > 1$, implying an unstable scheme, and thus, no convergence.

Here, $\parallel \boldsymbol{A} \parallel_\infty = \max\limits_{\boldsymbol{i}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} |\boldsymbol{a_{i\,j}}|$ is the infinite-norm of $\boldsymbol{A}$.

The stability condition $\parallel A \parallel_\infty \leq 1$ implies that:

(a) if $\sigma^2 > r$, then $\Delta_t < \frac{1}{\frac{r}{2}+\sigma^2 N^2}$,

(b) if $\sigma^2 < r$, then nothing can be said.

Comments: Given an arbitrary norm $\parallel . \parallel$, we can define stability as follows:

(a) A is strictly stable if $\parallel A^k \parallel \leq 1$ for any $k > 0$.

(b) A is strongly stable if $\parallel A^k \parallel \leq C$ for any $k > 0$ and constant C.

(c) If A is unstable then small errors can be amplified rendering the numerical scheme useless.

It will be seen later that stability will determine the convergence of the scheme for the EFD method.

Implicit Finite Difference Method (IFD)

Here we will use the Forward Difference Method for the time-derivative in approximating the PDE. The PDE for the transformed price, $X = Ln(S)$, is:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 c}{\partial x^2} + \nu \frac{\partial c}{\partial x} - rc = 0$$

When we use the forward difference for the time-derivative above, we obtain the following scheme:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta_t} + \frac{1}{2}\sigma^2 \cdot \frac{C_{i,j+1} - 2C_{i,j} + C_{i,j-1}}{\Delta x^2} + \nu \cdot \frac{C_{i,j+1} - C_{i,j-1}}{2\Delta x} - rC_{i,j} = 0$$

which can be rewritten as

$$C_{i+1,j} = P_u \cdot C_{i,j+1} + P_m \cdot C_{i,j} + P_d \cdot C_{i,j-1}$$

with

$$\begin{cases} P_u = -\frac{1}{2}\Delta_t\left(\frac{\sigma^2}{\Delta x^2} + \frac{\nu}{\Delta x}\right) \\ P_m = 1 + \Delta_t \cdot \frac{\sigma^2}{\Delta x^2} + r\Delta_t \\ P_d = -\frac{1}{2}\Delta_t\left(\frac{\sigma^2}{\Delta x^2} - \frac{\nu}{\Delta x}\right) \end{cases}$$

To solve the above system of equations, we need boundary conditions. We will use the following boundary conditions:

<u>For call options</u>: $C_{i,N} = C_{i,N-1} + \left(S_{i,N} - S_{i,N-1}\right)$, $\quad C_{i,-N} = C_{i,-N+1}$

<u>For put options</u>: $\pi_{i,N} = \pi_{i,N-1}$; $\quad \pi_{i,-N} = \pi_{i,-N+1} - (S_{i,-N} - S_{i,-N+1})$

Consider a call option.

For a fixed i, the above equations can be combined into a system of equations as follows:

$$\begin{cases} C_{i,N} - C_{i,N-1} = S_{i,N} - S_{i,N-1} \\ P_u \cdot C_{i,N} + P_m \cdot C_{i,N-1} + P_d \cdot C_{i,N-2} = C_{i+1,N-1} \\ P_u \cdot C_{i,N-1} + P_m \cdot C_{i,N-2} + P_d \cdot C_{i,N-3} = C_{i+1,N-2} \\ \vdots \\ P_u \cdot C_{i,-N+2} + P_m \cdot C_{i,-N+1} + P_d \cdot C_{i,-N} = C_{i+1,-N+1} \\ C_{i,-N+1} - C_{i,-N} = 0 \end{cases}$$

which can be written in a matrix form: $AF_i = B_i$, where

$$A = \begin{pmatrix} 1 & -1 & 0 & \vdots & \vdots & \vdots & \vdots & 0 \\ P_u & P_m & P_d & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & P_u & P_m & P_d & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & P_u & P_m & P_d & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & \vdots & P_u & P_m & P_d \\ 0 & 0 & \vdots & \vdots & \vdots & \vdots & 1 & -1 \end{pmatrix}, \quad F_i = \begin{pmatrix} C_{i,N} \\ C_{i,N-1} \\ C_{i,N-2} \\ C_{i,N-3} \\ . \\ . \\ C_{i,-N+1} \\ C_{i,-N} \end{pmatrix}, \quad B_i = \begin{pmatrix} S_{i,N} - S_{i,N-1} \\ C_{i+1,N-1} \\ C_{i+1,N-2} \\ C_{i+1,N-3} \\ . \\ . \\ C_{i+1,-N+1} \\ 0 \end{pmatrix}$$

The goal here is to find F_0.

Notice that F_i can be implicitly found from $AF_i = B_i$. To find the vector of option prices at time i, this matrix equation can be solved very efficiently using numerical methods of linear algebra for solving linear systems of equations. The accuracy of this method is $O(\Delta x^2 + \Delta_t)$, and more importantly, it is <u>unconditionally stable and convergent.</u>

So, while it has the same order of accuracy as the explicit finite difference method, we have more freedom to trade-off accuracy for speed by decreasing the time steps because we do not have to worry about a stability and convergence condition here.

Note that, we can no longer interpret P_u, P_m and P_d directly as probabilities, because P_u and P_d will typically be negative and P_m will be greater than one.

Crank-Nicolson Finite Difference Method

The Crank-Nicolson method is a refinement of the implicit finite difference method. It is a so-called fully centered method, because it replaces the space and time derivatives with finite differences centered at an imaginary time step at $(\boldsymbol{i} + \frac{\mathbf{1}}{\mathbf{2}})$. If we follow this idea, we will obtain the following finite difference equation:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta_t} + \frac{1}{2}\sigma^2 \left(\frac{(C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j-1}) + (C_{i,j+1} - 2C_{i,j} + C_{i,j-1})}{2\Delta x^2} \right)$$

$$+v \left(\frac{(C_{i+1,j+1} - C_{i+1,j-1}) + (C_{i,j+1} - C_{i,j-1})}{4\Delta x} \right) - r \left(\frac{C_{i+1,j} + C_{i,j}}{2} \right) = 0$$

which can be rewritten as

$$p_u C_{i,j+1} + p_m C_{i,j} + p_d C_{i,j-1} = -p_u C_{i+1,j+1} - (p_m - 2) C_{i+1,j} - p_d C_{i+1,j-1}$$

Consider call options.

Using the boundary conditions for call options, along with the above equation, we arrive at the following system of equations:

$$\begin{cases} C_{i,N} - C_{i,N-1} = z_{i+1,N} \\ P_u \cdot C_{i,N} + P_m \cdot C_{i,N-1} + P_d \cdot C_{i,N-2} = z_{i+1,N-1} \\ P_u \cdot C_{i,N-1} + P_m \cdot C_{i,N-2} + P_d \cdot C_{i,N-3} = z_{i+1,N-2} \\ \vdots \\ P_u \cdot C_{i,-N+3} + P_m \cdot C_{i,-N+2} + P_d \cdot C_{i,-N+1} = z_{i+1,-N+1} \\ C_{i,-N+1} - C_{i,-N} = z_{i+1,-N} = 0 \end{cases}$$

Here

$$z_{i+1,N} = S_{i,N} - S_{i,N-1}$$

$$z_{i+1,N-1} = -P_u \cdot C_{i+1,N} - (P_m - 2) C_{i+1,N-1} - P_d \cdot C_{i+1,N-2}$$

$$z_{i+1,N-2} = -P_u \cdot C_{i+1,N-1} - (P_m - 2) C_{i+1,N-2} - P_d \cdot C_{i+1,N-3}$$

$$\vdots$$

$$z_{i+1,-N+1} = -P_u \cdot C_{i+1,-N+2} - (P_m - 2) C_{i+1,-N+1} - P_d \cdot C_{i+1,-N}$$

$$z_{i+1,-N} = 0$$

These equations, along with the boundary conditions (discussed earlier) make up a tridiagonal system of equations similar to the implicit finite difference method, which can be written in a matrix form as follows:

$$\begin{pmatrix} 1 & -1 & 0 & 0 & \cdot & \cdot & 0 \\ P_u & P_m & P_d & 0 & \cdot & \cdot & 0 \\ 0 & P_u & P_m & P_d & \cdot & \cdot & 0 \\ . & . & . & . & \cdot & \cdot & . \\ . & . & . & . & \cdot & \cdot & . \\ . & . & . & . & \cdot & \cdot & . \\ 0 & 0 & 0 & 0 & P_u & P_m & P_d \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} C_{i,N} \\ C_{i,N-1} \\ C_{i,N-2} \\ . \\ . \\ . \\ C_{i,-N+1} \\ C_{i,-N} \end{pmatrix} = \begin{pmatrix} z_{i+1,N} \\ z_{i+1,N-1} \\ z_{i+1,N-2} \\ z_{i+1,N-3} \\ . \\ . \\ z_{i+1,-N+1} \\ z_{i+1,-N} \end{pmatrix}$$

$$\begin{cases} P_u = -\frac{1}{4}\Delta_t\left(\frac{\sigma^2}{\Delta x^2} + \frac{v}{\Delta x}\right) \\ P_m = 1 + \Delta_t \frac{\sigma^2}{2\Delta x^2} + \frac{r\Delta_t}{2} \\ P_d = -\frac{1}{4}\Delta_t\left(\frac{\sigma^2}{\Delta x^2} - \frac{v}{\Delta x}\right) \end{cases}$$

This system of equations can be solved very efficiently. The accuracy of this method is $O\left(\Delta x^2 + \left(\frac{\Delta_t}{2}\right)^2\right)$ and it is <u>unconditionally stable and convergent</u>. The Crank-Nicolson method converges faster than the implicit or the explicit finite difference methods.

As in the case of the implicit finite difference method, we can no longer interpret P_u, P_m and P_d directly as probabilities, because P_u and P_d will typically be negative and P_m will be greater than one.

<u>Another approach to numerically solve the PDE</u>

Consider the Black-Scholes PDE with boundary conditions (for call options):

$$C_t + \frac{1}{2}\sigma^2 S^2 C_{SS} + rS \cdot C_S - rC = 0$$

$C(0,t) = 0$, for all $0 < t < T$, $C(S,t) = S$ for $S \uparrow \infty$, and $C(S,T) = (S-K)^+$.

We will transform the PDE by setting $x = \ln\left(\frac{S}{K}\right)$, $\tau = \frac{1}{2}\sigma^2 \cdot (T-t)$, $C = K \cdot g(x,\tau)$.

Then, we have the following PDE in terms of new variables:

$$\begin{cases} g_\tau = g_{xx} + (\nu - 1) \cdot g_x - \nu \cdot g, \quad \nu = \frac{r}{\frac{1}{2}\sigma^2} \\ g(x,0) = (e^x - 1)^+ \end{cases}$$

Try $g(x,\tau) = e^{\alpha x + \beta\tau} \cdot f(x,\tau)$ (α, β are free parameters) as a solution of the above PDE. Choose the parameters α, β strategically, so that the coefficients in front of f and f_x are 0.

$$\beta \cdot f + f_\tau = \alpha^2 f + 2\alpha f_x + f_{xx} + (\nu - 1)(\alpha f + f_x) - \nu \cdot f$$

$$f(\beta - \alpha^2 - \alpha(\nu - 1) + \nu) + f_\tau = 2\alpha f_x + f_{xx} + (\nu - 1) f_x$$

That is, take $\beta = \alpha^2 + (\nu - 1)\alpha - \nu$, (the coefficient of $f = 0$), and $2\alpha + (\nu - 1) = 0$, (the coefficient of $f_x = 0$). Then, we get the solutions: $\hat{\alpha} = -\frac{1}{2}(\nu - 1)$, $\widehat{\beta} = -\frac{1}{4}(\nu + 1)^2$.

This gives the following functional form: $g(x,\tau) = e^{-\frac{1}{2}(\nu-1)x - \frac{1}{4}(\nu+1)^2 \cdot \tau} \cdot f(x,\tau)$.

Using it we arrive at the following simpler PDE:

$$f_\tau = f_{xx} \text{ with } f(x,0) = (e^{\frac{1}{2}(\nu+1)x} - e^{\frac{1}{2}(\nu-1)x})^+.$$

This is the well-known **Heat Equation**, which is not difficult to solve in a closed-form. However, we will consider its numerical solution, by using the finite difference methods.

Discretize this Heat equation to get

$$\frac{f_{i+1,j} - f_{i,j}}{\Delta_t} = \frac{f_{i,j+1} - 2 \cdot f_{i,j} + f_{i,j-1}}{(\Delta x)^2}$$

$$f_{i+1,j} - f_{i,j} = \alpha \cdot f_{i,j+1} - 2 \cdot \alpha f_{i,j} + \alpha \cdot f_{i,j-1}$$

$$f_{i+1,j} = \alpha \cdot f_{i,j+1} + (1 - 2\alpha) \cdot f_{i,j} + \alpha \cdot f_{i,j-1}$$

with $\alpha = \frac{\Delta_t}{(\Delta x)^2}$.

Comment: In order for the scheme to converge, we must have $\alpha = \frac{\Delta_t}{(\Delta x)^2} \leq \frac{1}{2}$.

How to choose the ranges for the discretization parameters?

For x: Take x to be between $-N(\Delta x)$ and $+N(\Delta x)$: $-N\Delta x \leq x \leq N\Delta x$

For τ: Take $\Delta\tau = \frac{1}{2}\sigma^2 T/M$ and divide $\left[0, \frac{1}{2}\sigma^2 T\right]$ into M equal parts.

Then, $0 = \tau_0 < \tau_1 < \cdots < \tau_M = \frac{1}{2}\sigma^2 T$

$$\begin{cases} f_{i,-N} \text{ are given by boundary conditions } \forall\, i = 0, \dots, M \\ f_{i,N} \text{ are given by boundary conditions } \forall\, i = 0, \dots, M \\ f_{0,j} \text{ are given by boundary conditions } \forall\, j = -N, \dots, N \end{cases}$$

The goal: Find $f_{i,j}$ for $i = M,\ j =$all values.

The accuracy of this method is $O(\Delta x^2 + \Delta_t)$ which means that if we halve $\Delta x^2 + \Delta_t$, then we can halve the error. Therefore, to halve the error we must halve the time step, but only need to reduce the space step by a factor of $1/\sqrt{2}$.

Stability and Convergence

For the Explicit Finite Difference method, it is important to ensure that the numbers $\boldsymbol{P_u}$, $\boldsymbol{P_m}$ and $\boldsymbol{P_d}$ are positive and that the stability and convergence condition $\boldsymbol{\Delta x \geq \sigma\sqrt{3\Delta_t}}$ is satisfied.

The discretization error can be reduced by using smaller space and time steps. A finite difference method is convergent if the discretization error tends to zero as the space and time steps tend to zero, and it is stable if the round-off error is small and remains bounded for all time.

There are two fundamental sources of error in our schemes: the truncation error in the space, and the discretization error.

The truncation error may cause problems in that the numerical scheme may solve a problem, but not exactly the problem we are trying to solve (say the Black-Scholes PDE).

When solving PDEs numerically, there are three important issues to take into consideration: consistency, stability, and convergence of schemes.

1. **CONSISTENCY** A numerical scheme is consistent if the finite difference scheme converges to the PDE, as the time- and space-steps tend to zero. (This could be a problem when they converge to zero NOT in separation).
2. **STABILITY** A numerical scheme is stable if the difference between the numerical solution and the exact solution remains bounded as the step sizes converge to zero.
3. **CONVERGENCE** A numerical scheme converges if the difference between numerical solution and exact solution converges to zero at any point of the domain, as the time- and space-steps tend to zero.

LEMMA: The LAX Equivalence Theorem. Given a properly posed linear initial value problem and a consistent finite difference scheme, **stability is sufficient for the convergence** of the scheme.

The Lemma implies that if the scheme is not stable, small errors are amplified and result in divergence of the scheme.

Method	**Stability**	**Accuracy**	**Comments**
EFD	Not guaranteed. Stable iff $\parallel A \parallel_{\infty} \leq 1$	$O(\Delta x^2 + \Delta_t)$	May diverge if parameters not properly chosen. Choose:

			(a) $\Delta_t \leq \frac{1}{3}\left(\frac{\Delta x}{\sigma}\right)^2$ for stability (in case of X-PDE), (b) $\Delta_t \leq \frac{1}{2}(\Delta S)^2$ for stability (in case of S-PDE).
IFD	unconditionally stable	$O(\Delta x^2 + \Delta_t)$	unconditionally convergent
C-NFD	unconditionally stable	$O\left(\Delta x^2 + \left(\frac{\Delta_t}{2}\right)^2\right)$	unconditionally convergent

Consider one of the discretization methods. If the volatility becomes small relative to the interest rate, then P_d can become negative. Furthermore, if the volatility becomes large, then the condition $\Delta x \geq \sigma\sqrt{3\Delta_t}$ can be violated and this can lead to p_m becoming negative and resulting divergence of the system, although this is not the only reason for instability and lack of convergence.

Let us consider finding a reasonable number of time steps to be used with a typical volatility of 25% ($\sigma = 0.25$).

A reasonable range of asset price values at the maturity date of the option is 3 standard deviations either side of the mean and a reasonable number of asset price values is 100 ($2N_j + 1 = 100$). This allows us to calculate the space step required:

$$\Delta x = \frac{6\sigma\sqrt{T}}{100} = 0.015$$

If we now apply the condition $\Delta x \geq \sigma\sqrt{3\Delta_t}$, then we obtain

$$\Delta_t \leq \frac{1}{3}\left(\frac{\Delta x}{\sigma}\right)^2 = 0.0012$$

This leads to a requirement of more than 833 time-steps per year.

Generalization of Finite-Difference Schemes

Take a grid node (i, j) which corresponds to $t_i = i \cdot \Delta t,$ and $S_j = j \cdot \Delta S$. The option value in that node will be denoted by $C_{i,j}$. The partial derivative $\frac{\partial C}{\partial t}$ of C with respect to time at node (i, j) can be approximated as follows:

$$\frac{\partial C}{\partial t} \approx \frac{C_{i+1,j} - C_{i,j}}{\Delta t}$$

For other derivatives, such as $\frac{\partial C}{\partial S}, \frac{\partial^2 C}{\partial S^2}$, we will take a linear combination of finite-difference approximation <u>at two neighboring nodes to (i, j):</u>

$$\frac{\partial C}{\partial S} \approx \alpha \cdot \left(\frac{C_{i+1,j+1} - C_{i+1,j-1}}{2 \cdot \Delta S}\right) + (1 - \alpha) \cdot \left(\frac{C_{i,j+1} - C_{i,j-1}}{2 \cdot \Delta S}\right)$$

$$\frac{\partial^2 C}{\partial S^2} \approx \alpha \cdot \left(\frac{C_{i,j+1} - 2C_{i,j} + C_{i,j-1}}{(\Delta S)^2}\right) + (1 - \alpha) \cdot \left(\frac{C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j-1}}{(\Delta S)^2}\right)$$

Also, for C we will use $C \approx \alpha \cdot C_{i+1,j} + (1 - \alpha)C_{i,j}$

The F.D. approximation of Black-Scholes PDE, $\frac{\partial C}{\partial t} + r \cdot S \cdot \frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 \cdot S^2 \cdot \frac{\partial^2 C}{\partial S^2} - r \cdot C = 0$, will result in following scheme:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta} + \frac{r \cdot j}{2} \cdot \left[\alpha C_{i+1,j+1} - \alpha C_{i+1,j-1}\right] + \frac{r \cdot j}{2} \cdot \left[(1 - \alpha)C_{i,j+1} - (1 - \alpha)C_{i,j-1}\right]$$
$$+ \frac{1}{2}\sigma^2 j^2 \left[\alpha \cdot C_{i+1,j+1} - 2 \cdot \alpha \cdot C_{i+1,j} + \alpha \cdot C_{i+1,j-1}\right]$$
$$+ \frac{1}{2}\sigma^2 j^2 \left[(1 - \alpha)C_{i,j+1} - 2(1 - \alpha)C_{i,j} + (1 - \alpha)C_{i,j-1}\right] - r \cdot \alpha \cdot C_{i+1,j} - r(1 - \alpha) \cdot C_{i,j}$$
$$= 0$$

By combining similar terms, we can rewrite it as:

$$C_{i,j-1}\cdot\left\{\frac{-(1-\alpha)\cdot r\cdot j}{2}+\frac{\sigma^2 j^2\cdot(1-\alpha)}{2}\right\}+C_{i,j}\left\{-\frac{1}{\Delta}-\sigma^2 j^2\cdot(1-\alpha)-r(1-\alpha)\right\}+C_{i,j+1}$$

$$\cdot\left\{\frac{(1-\alpha)\cdot r\cdot d}{2}+(1-\alpha)\cdot\frac{1}{2}\sigma^2\cdot j^2\right\}+C_{i+1,j-1}\left\{-\frac{\alpha r\cdot j}{2}+\alpha\cdot\frac{1}{2}\sigma^2\cdot j^2\right\}+C_{i+1,j}$$

$$\cdot\left\{\frac{1}{\Delta}-\alpha\sigma^2 j^2-r\cdot\alpha\right\}+C_{i+1,j+1}\cdot\left\{\frac{\alpha r\cdot j}{2}+\alpha\cdot\frac{1}{2}\sigma^2\cdot j^2\right\}=0$$

Denote

$$a_1=-\frac{r\cdot j(1-\alpha)}{2}+\frac{\sigma^2\cdot j^2(1-\alpha)}{2}=\frac{(\sigma^2 j^2-r\cdot j)\cdot(1-\alpha)}{2}$$

$$a_2=-\frac{1}{\Delta}-\sigma^2 j^2\cdot(1-\alpha)-r(1-\alpha)=-\frac{1}{\Delta}-(\sigma^2 j^2+r)(1-\alpha)$$

$$a_3=\frac{r\cdot j(1-\alpha)}{2}+\frac{\sigma^2 j^2(1-\alpha)}{2}=\frac{(\sigma^2 j^2+r\cdot j)\cdot(1-\alpha)}{2}$$

$$b_1=\frac{-r\cdot j\cdot\alpha}{2}+\frac{\sigma^2 j^2\cdot\alpha}{2}=\frac{(\sigma^2 j^2-rj)\cdot\alpha}{2}$$

$$b_2=\frac{1}{\Delta}-\sigma^2 j^2\cdot\alpha-r\cdot\alpha=\frac{1}{\Delta}-(\sigma^2 j^2+r)\cdot\alpha$$

$$b_3=\frac{rj\cdot\alpha}{2}+\frac{\sigma^2 j^2\cdot\alpha}{2}=\frac{(\sigma^2 j^2+r\cdot j)\cdot\alpha}{2}$$

<u>Note</u>: The coefficients $a_1, a_2, a_3, b_1, b_2, b_3$ are j-dependent, but we omit that index for simplicity of notations.

Then, we can write the following scheme:

$$a_1\cdot C_{i,j-1}+a_2\cdot C_{i,j}+a_3\cdot C_{i,j+1}+b_1\cdot C_{i+1,j-1}+b_2\cdot C_{i+1,j}+b_3\cdot C_{i+1,j+1}=0$$

Solving this equation backwards in time will result in option prices at time i, given the values of the option are known at time $i+1$.

$$a_1 \cdot C_{i,j-1} + a_2 \cdot C_{i,j} + a_3 \cdot C_{i,j+1} = d_{i+1,j}$$

where $d_{i+1,j} = -b_1 \cdot C_{i+1,j-1} - b_2 \cdot C_{i+1,j} - b_3 \cdot C_{i+1,j+1}$.

Using option values in 3 nodes at time $i + 1$ will help us find option values at 3 nodes at time i.

By combining the above equations in a matrix equation, we can solve for vector of option prices, as was done in the three cases discussed earlier.

prices, as was done in the three cases discussed earlier.

$AC_i = B_{i+1}$, where

$$A = \begin{pmatrix} a_2^{N-1} & a_1^{N-1} & 0 & \vdots & \vdots & \vdots & \vdots & 0 \\ a_3^{N-2} & a_2^{N-2} & a_1^{N-2} & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & a_3^{N-3} & a_2^{N-3} & a_1^{N-3} & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & a_3^{j} & a_2^{j} & a_1^{j} & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & 0 & a_3^2 & a_2^2 & a_1^2 \\ 0 & 0 & \vdots & \vdots & 0 & 0 & a_3^1 & a_2^1 \end{pmatrix}, \quad C_i = \begin{pmatrix} C_{i,N-1} \\ C_{i,N-1} \\ C_{i,N-2} \\ . \\ . \\ C_{i,2} \\ C_{i,1} \end{pmatrix}, \quad B_{i+1} = \begin{pmatrix} d_{i+1,\ N-1} \\ d_{i+1,\ N-2} \\ d_{i+1,\ N-3} \\ . \\ . \\ . \\ d_{i+1,\ 1} \end{pmatrix}$$

The goal here is to find C_0.

The above matrix equation can easily be solved to estimate C_i.

Remark: We did not include the extreme values of C in the calculations above. That is, the values of $C_{i,N}$ and $C_{i,0}$ that correspond to option values for "very large" and "very small" stock prices, respectively, were not included in the matrix equation above.

- For call options, we can use $C_{i,0} = C_{i,1}$ and $C_{i,N} = C_{i,N-1} + \left(S_{i,N} - S_{i,N-1}\right)$, for $i = 0, \ldots, M$.
- For put options, we can use $\pi_{i,0} = \pi_{i,1} - \left(S_{i,0} - S_{i,1}\right), and\ \ \pi_{i,N} = \pi_{i,N-1}$ for $i = 0, \ldots, M$.

Special Cases:

(1) $\alpha = 1$. This will be the **explicit** F.D. method

(2) $\alpha \neq 1$. This is an **implicit** F.D. method.

(3) $\alpha = 0$. This is the **fully implicit** F.D. method.

(4) $\alpha = \frac{1}{2}$. This is the **Crank-Nicolson** F.D. method.

Classification of PDEs

Consider the following PDE

$$\alpha(x,y)\frac{\partial f}{\partial x} + b(x,y)\frac{\partial f}{\partial y} + c(x,y) \cdot f + d(x,y) = 0$$

This is a first-order PDE since only first order derivatives are involved in the equation. Second order linear PDEs are given by

$$\alpha(x,y) \cdot \frac{\partial f}{\partial x^2} + b(x,y) \cdot \frac{\partial^2 f}{\partial x \partial y} + c(x,y) \cdot \frac{\partial^2 f}{\partial y^2} + d(x,y) \cdot \frac{\partial f}{\partial x} + e(x,y) \cdot \frac{\partial f}{\partial y} + g(x,y) \cdot f(x,y) = 0$$

PDEs are classified as follows:

- Hyperbolic if $b^2 - 4ac > 0$
- Parabolic if $b^2 - 4ac = 0$
- Elliptic if $b^2 - 4ac < 0$

For example, for the B.S. PDE is $\frac{\partial C}{\partial t} + r \cdot S \cdot \frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 \cdot S^2 \cdot \frac{\partial^2 C}{\partial S^2} - r \cdot C = 0$. $b^2 - 4ac = 0$, therefore, it is a parabolic PDE.

$$\underbrace{\left(\frac{1}{2}\sigma^2 \cdot S^2\right)}_{a} \cdot \frac{\partial^2 C}{\partial S^2} + \underbrace{0}_{b} \cdot \frac{\partial^2 C}{\partial t \partial S} + \underbrace{0}_{c} \cdot \frac{\partial^2 C}{\partial t^2} + \underbrace{(r \cdot S)}_{d} \cdot \frac{\partial C}{\partial S} + \underbrace{1}_{e} \cdot \frac{\partial C}{\partial t} + r \cdot C = 0$$

The Feyman-Kac Theorem

Below we provide the relationship between two numerical methods of estimating derivative security prices – the PDE approach and the probabilistic (Monte Carlo) method.

Theorem (Feyman-Kac).

(a) Assume the price-process is given by $dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dW_t$ under the risk-neutral measure, and that the price of a contingent claim at time T is given by

$$V(t, S_t) = \mathbb{E}_t^* \left(e^{-\int_t^T r(u, S_u)du} \cdot H(T, S_T) \right)$$

Then,

$$\frac{\partial V}{\partial t} + \mu \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial S^2} - rV = 0 \quad \text{with } V(T, S_T) = H(T, S_T)$$

(b) In a multidimensional case when, $dS_t^i = \mu_i(t, S_t^i)dt + \sigma_i(t, S_t^i)dW_t^i$ for $i = 1, \dots, d$, the PDE is given by:

$$\frac{\partial V}{\partial t} + \sum_{i=1}^{d} \mu_i \frac{\partial V}{\partial S_i} + \frac{1}{2} \sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} \frac{\partial^2 V}{\partial S_i \partial S_j} - rV = 0$$

(c) Assume that the price-process is given by $dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dW_t$ under the risk-neutral measure , and the price of a contingent claim at time T is given by

$$V(t, S_t) = \mathbb{E}_t^* \left(\int_t^T h(u, S_u)\, e^{-\int_t^u r(y, S_y)dy} du + e^{-\int_t^T r(y, S_y)dy} \cdot g(T, S_T) \right)$$

Then, $\frac{\partial V}{\partial t} + \mu \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial S^2} - rV + h = 0$ with $V(T, S_T) = g(T, S_T)$.

Example. **Implementation of the IFD method in pricing a European Call Option on a Pure Discount Bond in the CIR framework.**

Consider the CIR model of short-term rate: $dr_t = \kappa(\bar{r} - r_t)dt + \sigma\sqrt{r_t}dW_t$.

The goal is to compute the price $C(0,T,S)$ of the European Call option on a pure discount bond (zero-coupon bond), which matures at time *S* and pays *FV* at maturity. The option's expiration is at time T (where $S > T$), and the strike price of the option is *K*.

Under the risk-neutral measure, the price of the call option, at time 0, is given by:

$$C(0,T,S) = \mathbb{E}\left(e^{-\int_0^T r_s ds}\ \ (\boldsymbol{P}(T,S,r_T)\ - K)^+\ \ \right)$$

where $\boldsymbol{P}(T,S,r_T)$ is the price of the pure discount bond at time T, maturing at time S and paying FV at maturity.

Using the Feyman-Kac Theorem, we can transform the problem into another one in which we solve the following PDE for C

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{t}} + \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\sigma}^{\mathbf{2}}\boldsymbol{r}\frac{\partial^{\mathbf{2}}\boldsymbol{C}}{\partial \boldsymbol{r}^{\mathbf{2}}} + \boldsymbol{\kappa}(\bar{\boldsymbol{r}} - \boldsymbol{r})\ \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{r}} - \boldsymbol{rC} = \mathbf{0}$$

With the terminal condition, $C(T,T,S) = \max(\boldsymbol{P}(T,S,r_T)\ - K,\ 0)$, where $\boldsymbol{P}(T,S,r_T)$ is the price of the pure discount bond at time T, maturing at time S and paying FV at maturity.

While there is a closed-form solution to the above-posed problem, below we provide the details of an algorithm for numerically solving the above PDE, by using the Implicit Finite Difference (IFD) method.

Consider a grid of time (t) and rate (r) as follows:

Take a uniform partition of the time interval [0,T] by dividing the time-interval [0,T] into M equal parts: $0 = t_0 < t_1 < \cdots < t_{M-1} < t_M = T$, where $t_k = \frac{T}{M}k = \Delta k$, where $\Delta = \frac{T}{M}$.

Take a truncated range of r as follows: $[0, r_{max}]$, where $r_{max} = N\Delta r$, r takes on the following values $\{0, \Delta r, 2\Delta r, \dots, N\Delta r\}$. That is, $r_{i,j} = j\Delta r$, for any $i = 0, \dots, M$ $j = 0, \dots, N$.

<u>The boundary conditions/values of option prices can be used as follows:</u>

(4) $C_{M,j} = (P_{M,j} - K)^+$ for $j = 0, \dots, N$ (this is the payoff condition at option's expiration)

(5) $C_{i,N} = 0$, for $i = 0, \dots, M$ (since for very large values of r, the call option is deep out of the money, and the call value is worthless)

(6) $C_{i,0} = PV - K$, for $i = 0, \dots, M$ ($\text{since for small r, r} = 0, \text{the option is deep in} - \text{the} - \text{money and its value is } PV - K.$

The discretized version of this PDE (using the Implicit FD method) can be written as follows:

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta t} + \frac{1}{2}\sigma^2(j\Delta r)\frac{C_{i,j+1} - 2C_{i,j} + C_{i,j-1}}{(\Delta r)^2} + \kappa(\bar{r} - j\Delta r)\frac{C_{i,j+1} - C_{i,j-1}}{2\Delta r} - rC_{i,j} = 0$$

By combining the similar terms, we can rewrite the above scheme as:

$$C_{i+1,j} = P_u^j \cdot C_{i,j+1} + P_m^j \cdot C_{i,j} + P_d^j \cdot C_{i,j-1}$$

where

$$\begin{cases} P_u^j = \Delta_t\left(-\dfrac{\sigma^2 j}{2\Delta r} - \dfrac{\kappa\bar{r}}{2\Delta r} + \dfrac{\kappa j}{2}\right) \\ P_m^j = \Delta_t\left(\dfrac{1}{\Delta_t} + \dfrac{\sigma^2 j}{\Delta r} + r\right) \\ P_d^j = \Delta_t\left(-\dfrac{\sigma^2 j}{2\Delta r} + \dfrac{\kappa\bar{r}}{2\Delta r} - \dfrac{\kappa j}{2}\right) \end{cases}$$

for $j = N - 1, \dots, 1$.

Also, we have the boundary conditions for C: $C_{i,0} = PV - K$ and $C_{i,N} = 0$, for $i = 0, \dots, M$.

Putting all equations together, we will have the following system of equations in a matrix form:

$$C_{i+1} = AC_i + B_i$$

where

$$C_i = \begin{pmatrix} C_{i,N-1} \\ C_{i,N-2} \\ C_{i,N-3} \\ . \\ . \\ C_{i,1} \end{pmatrix}; \quad A = \begin{pmatrix} P_m^{N-1} & P_d^{N-1} & 0 & \vdots & \vdots & \vdots & \vdots & 0 \\ P_u^{N-2} & P_m^{N-2} & P_d^{N-2} & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & P_u^{N-3} & P_m^{N-3} & P_d^{N-3} & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & P_u^{N-4} & P_m^{N-4} & P_d^{N-4} & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & 0 & P_u^2 & P_m^2 & P_d^2 \\ 0 & 0 & \vdots & \vdots & 0 & 0 & P_u^1 & P_m^1 \end{pmatrix}; \quad B_i = \begin{pmatrix} P_u^N C_{i,N} \\ 0 \\ 0 \\ 0 \\ . \\ . \\ P_d^1 C_{i,0} \end{pmatrix}$$

The goal is to find C_0.

The matrix equations $C_{i+1} = AC_i + B_i$ can be solved, starting at $i = N - 1$ and moving backwards in time to $i = 0$.

Remarks:

1. We did not include the extreme values of *C* in the matrix equations above. That is, the values of $C_{i,N}$ and $C_{i,0}$, which correspond to option values for “very large” and “very small” bond prices, respectively, were not included in the matrix equation above.
2. One obtains a “very large” (or small) bond price when the rate is 0 (or r_{max}).
3. For call options, and for $i = 0, \dots, M$ we can use $C_{i,0} = PV - K$ as this is the case when *r=0* and the call option is deep in-the-money. Also, $C_{i,N} = 0$, for $i = 0, \dots, M$. This corresponds to the case when $r = r_{max}$, when the underlying bond price is very low, so the option is close-to-being worthless.

Exercises:

1. Consider the following situation on the stock of company XYZ: The current stock price is \$10, and the volatility of the stock price is $\sigma = 20\%$ per annum. Assume the prevailing risk-free rate is $r = 4\%$ per annum. Use the $X = ln(S)$ transformation of the Black-Scholes PDE, and $\Delta t = 0.002,$ with $\Delta X = \sigma\sqrt{\Delta t},$ or with $\Delta X = \sigma\sqrt{3\Delta t},$ or with $\Delta X = \sigma\sqrt{4\Delta t}$, and a *uniform* grid to price a European Put option with strike price of $K = \$10$, maturity of 0.5-years, and current stock prices for a range from \$4 to \$16; using the specified methods below:

    (a) *Explicit Finite-Difference method,*

    (b) *Implicit Finite-Difference method,*

    (c) *Crank-Nicolson Finite-Difference method.*

2. Consider the following situation on the stock of company XYZ: The current stock price is \$10, and the volatility of the stock price is $\sigma = 20\%$ per annum. Assume the prevailing risk-free rate is $r = 4\%$ per annum.

    Use the Black-Scholes PDE (for S) to price American Call and American Put options with strike prices of $K = \$10$, maturity of 0.5-years, and current stock prices for a range from \$4 to \$16; using the specified methods below:

    (a) *Explicit Finite-Difference method,*

    (b) *Implicit Finite-Difference Method,*

    (c) *Crank-Nicolson Finite-Difference method.*

    Choose $\Delta t = 0.002$, with $\Delta S = 0.5$, or with $\Delta S = 1$, or with $\Delta S = 1.5$.

3. Consider the following situation on the stock of company XYZ: The current stock price is \$10, and the volatility of the stock price is $\sigma = 20\%$ per annum. Assume the prevailing risk-free rate is $r = 4\%$ per annum. Use the original Black-Scholes PDE

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\ \frac{\partial P}{\partial S} - rP = 0$$

$\Delta t = 0.002$, $\mathrm{u} = \mathrm{e}^{\sigma\sqrt{0.25\,\Delta_t}}$, or $\mathrm{u} = \mathrm{e}^{\sigma\sqrt{\Delta_t}}$, or $\mathrm{u} = \mathrm{e}^{\sigma\sqrt{4\Delta_t}}$, and a *binomial/trinomial* grid to price the specified options using the specified methods below:

Use the *Explicit Finite-difference method*, the *Implicit Finite-Difference Method,* and *Crank-Nicolson Finite-Difference Method* to price a American Put option with strike price of $K = \$10$, maturity of 0.5-years, and current stock prices for a range from \$4 to \$16.

4. Consider the following situation on the stock of company XYZ: The current stock price is \$10, and the volatility of the stock price is $\sigma = 20\%$ per annum. Assume the prevailing risk-free rate is $r = 4\%$ per annum. Use the transformation of the Black-Scholes PDE to the **Heat Equation**, $\Delta t = 0.002$, $\Delta X = \sqrt{4\Delta_t}$, or $\Delta X = \sqrt{2\Delta_t}$, or $\Delta X = \sigma\sqrt{1.9\ \Delta_t}$, and a *uniform* grid to price the following options using the specified methods below:

Use the *Explicit Finite-difference method*, the *Implicit Finite-Difference Method,* and *Crank-Nicolson Finite-Difference Method* to price an American Call and Put option with strike price of $K = \$10$, maturity of 0.5-years, and current stock prices for a range from \$4 to \$16.