

Numerical methods in finance



유한차분법 (Finite Difference Methods)

Lecture 03

Fall 2025
KAIST MFE
금융수치해석기법

간단한 예제

Ordinary differential equation:

$$\frac{\partial f}{\partial x} = -\frac{f(x)}{x}$$

- 미분방정식의 해는 이 식을 만족하는 함수 $f(x)$ 를 찾는 것임
- 이 ODE의 해는 다음과 같이 구할 수 있음:
 $f(x) = C/x$
- 위의 해는 임의의 상수 C 를 포함 \rightarrow 무한히 많은 해를 가짐
- C 를 결정하기 위해서는 제약조건(constraint)가 필요 함
- Initial condition: 예를 들어 $f(2) = 1$ 라면 $C = 2$ 임을 알 수 있음

미분방정식 해법

분석적 해(Aalytical Solution) 찾기

- 문제를 완전히 해결할 수 있는 명확한 수학적 공식이나 해를 제공
- 대부분의 상황에서 해를 찾기가 어렵거나 해가 존재하지 않는 경우가 많음

수치적 근사법(Numerical Approximation):

- 수치적 방법은 주어진 유한 개의 점에서 미분 방정식을 만족하는 함수를 근사하여 모든 방정식의 해를 제공
- 함수 전체에 대한 직관적인 이해가 어려울 수 있음

수치적 근사 – explicit method

Forward approximation

$$\frac{\partial f}{\partial x} = -\frac{f(x)}{x} \quad \Rightarrow \quad \frac{f(x + \Delta x) - f(x)}{\Delta x} = -\frac{f(x)}{x}$$
$$f(x + \Delta x) = f(x) \left(1 - \frac{\Delta x}{x} \right)$$

Explicit method

- 알려진 다른 node의 값으로부터 grid상의 node에서 함수값을 직접 계산하는 방법
- $x_0 = 2$ 이고 $f(x_0) = 1$ 인 지점에서 시작해서 $x_0 + \Delta x$ 의 값을 수치적으로 계산
- $x_N = 10$ 일 때 계산 결과 ($\Delta x = (x_N - x_0)/N$)

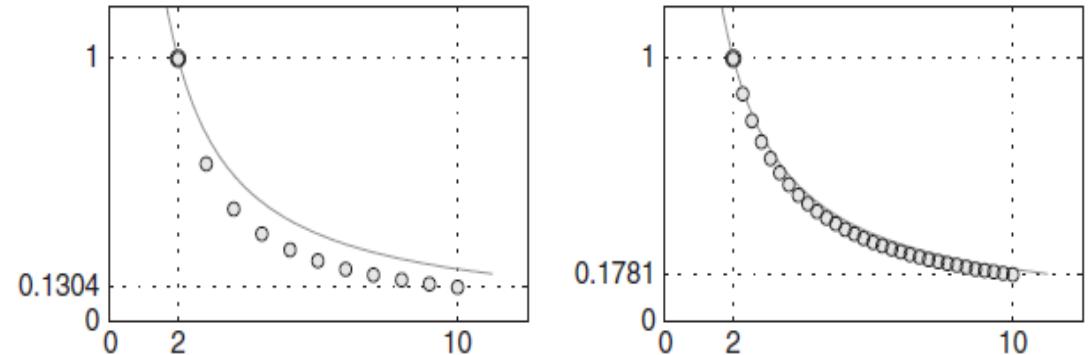


Figure 4.1 Numerical approximation with the explicit method (circles) of the differential equation defined in Eq. (4.1) for $N = 10$ (left panel) and $N = 30$ (right panel).

다른 근사 방법

Central difference

$$\frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} = -\frac{f(x)}{x}$$
$$f(x) = \frac{x}{2\Delta x} (f(x - \Delta x) - f(x + \Delta x))$$

- 이 방법을 적용하기 위해서는 2개의 점 ($f(x - \Delta x)$ 와 $f(x + \Delta x)$) 을 알아야 함

Implicit method

- 4개의 연속한 점에 대한 추정의 경우 다음과 같이 선형방정식으로 표현할 수 있음

$$\begin{aligned}y_1 &= \frac{x_1}{2\Delta x}(y_0 - y_2) \\y_2 &= \frac{x_2}{2\Delta x}(y_1 - y_3) \\y_3 &= \frac{x_3}{2\Delta x}(y_2 - y_4) \\y_4 &= \frac{x_4}{2\Delta x}(y_3 - y_5)\end{aligned}\implies \begin{bmatrix} 1 & c_1 & & & \\ -c_2 & 1 & c_2 & & \\ & -c_3 & 1 & c_3 & \\ & & -c_4 & 1 & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} c_1 y_0 \\ 0 \\ 0 \\ -c_4 y_5 \end{bmatrix}$$

- 여기에서 $c_i = x_i/(2\Delta x)$. 추가로 y_5 를 알아야 해를 구할 수 있음(final condition)
- 이 문제의 Final condition은 $x \rightarrow \infty$ 일 때, $f(x) \rightarrow 0$ 의 특성을 이용할 수 있음
- 충분히 큰 x 에 대해서 0으로 간주해도 문제없음

Implicit method

- 마지막 node 값인 x_N 을 충분히 큰 값으로 하고, $\Delta x = (x_N - x_0)/N$ 이라고 할 때 결과

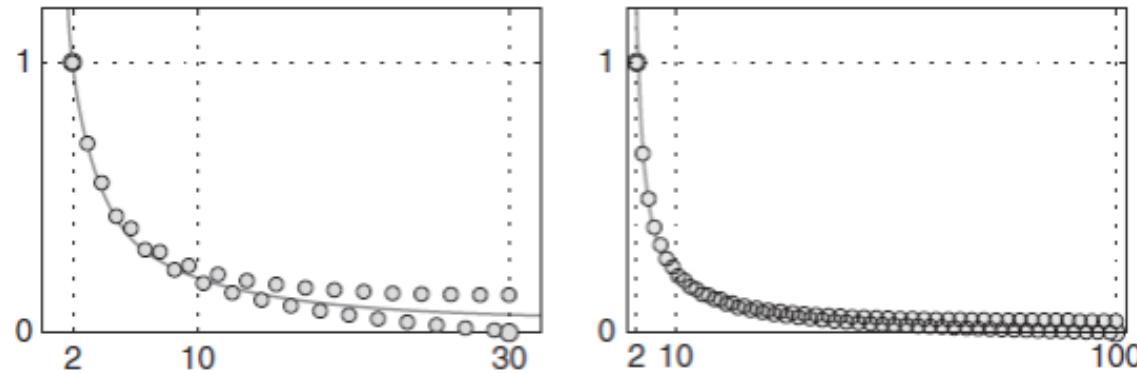


Figure 4.2 Numerical approximation with the implicit method (circles) of the differential equation defined in Eq. (4.1) for $x_N = 30$ and $N = 30$ (left panel) and $x_N = 100$ and $N = 80$ (right panel).

- 이상의 두 가지 해법에 따라서 계산 결과가 달라짐
- 또한 step length(Δx)와 final conditions은 근사 오차와 계산의 복잡도를 결정함
- Implicit method의 경우 step size가 작을수록 선형방정식 시스템의 크기가 커짐

미분방정식과 블랙-숄즈방정식

- $u(x, y)$ 는 x 와 y 에 대한 함수
- 편미분(partial derivatives) $u_x \equiv \frac{\partial u}{\partial x}$, $u_y \equiv \frac{\partial u}{\partial y}$
- ODE(Ordinary differential equations) – 단일 독립 변수를 가지는 함수
- PDE(Partial differential equations) – 2개 이상의 독립 변수를 가지는 함수

Black-Scholes Equation

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - q)S \frac{\partial V}{\partial S} - rV = 0 \iff V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - q)SV_S - rV = 0$$

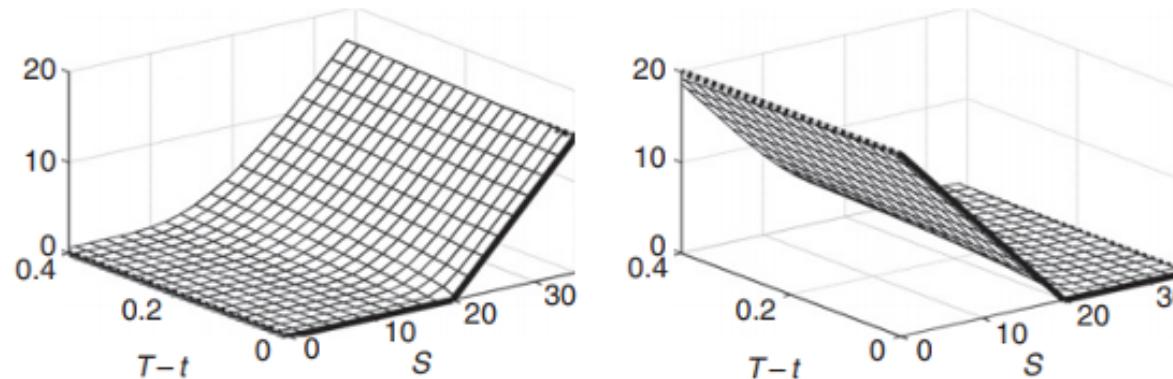


Figure 4.3 Price of call option (left panel) and price of put option (right panel) for $X = 20$, $r = 0.10$, $q = 0$ and $\sigma = 0.60$.

Initial and boundary conditions / definition of grid

Initial and boundary conditions for a European call option:

$$V(S, T) = \max(S - K, 0)$$

$$V(0, t) = 0, \quad \lim_{S \rightarrow \infty} V(S, t) = S - Ke^{-r(T-t)}$$

Initial and boundary conditions for a European put option:

$$V(S, T) = \max(K - S, 0)$$

$$V(0, t) = Ke^{-r(T-t)}, \quad \lim_{S \rightarrow \infty} V(S, t) = 0$$

Space domain: $[S_{min}, S_{max}]$

- N 등분 grid points, $\Delta S = (S_{max} - S_{min})/N$
- $S_i = S_{min} + i\Delta S, \quad i = 0, \dots, N$

Time domain: 0 to T in M steps

- $\Delta t = T/M$
- $t_j = j\Delta t, \quad j = 0, \dots, M$

Terminal and boundary conditions

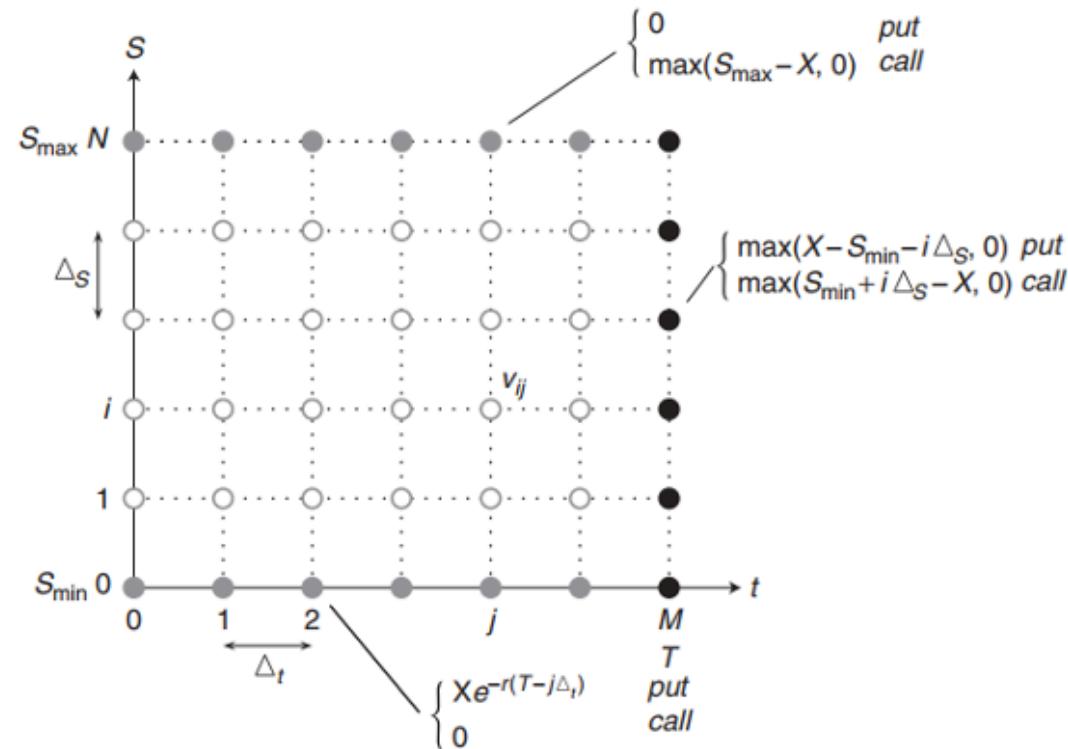


Figure 4.4 Finite difference grid with terminal conditions (dark circles) and boundary conditions (gray circles).

- Function evaluations at the grid points v_{ij}
- $v_{ij} = V(S_{\min} + i\Delta S, j\Delta t)$

Derivatives approximation

- The derivative V_t of the option with respect to time is approximated with a backward difference

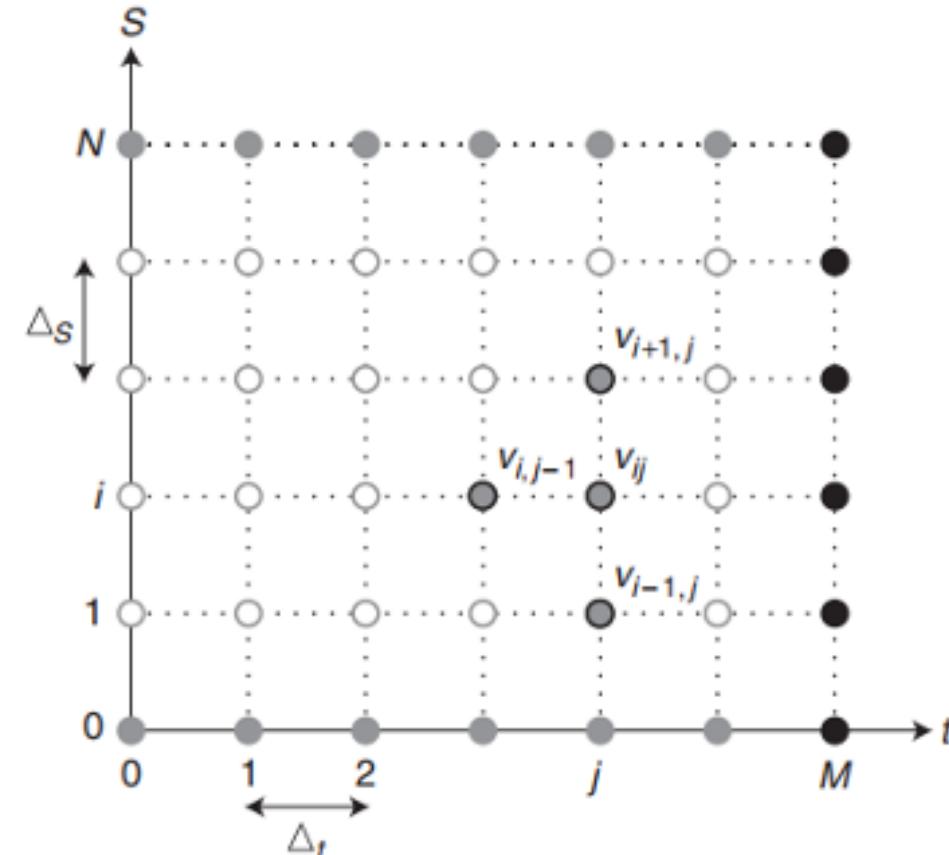
$$V_t|_{t=j\Delta t} \approx \frac{v_{i,j} - v_{i,j-1}}{\Delta t}$$

- The derivative of the option with respect to the stock price is approximated with a central difference

$$V_S|_{S=S_{min}+i\Delta S} \approx \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta S}$$

- The second-order derivative of the option with respect to the stock price

$$V_{SS}|_{S=S_{min}+i\Delta S} \approx \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta S)^2}$$



Black-Scholes equation with approximations

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - q)S V_S - rV = 0$$
$$\implies \frac{v_{i,j} - v_{i,j-1}}{\Delta t} + \frac{1}{2}\sigma^2 S_i^2 \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta S)^2} + (r - q)S_i \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta S} - rv_{i,j} = 0$$

Denoting $a_i = \Delta t \frac{\sigma^2 S_i^2}{2(\Delta S)^2}$ and $b_i = \Delta t \frac{(r-q)S_i}{2\Delta S}$,

$$v_{i,j} - v_{i,j-1} + a_i(v_{i+1,j} - 2v_{i,j} + v_{i-1,j}) + b_i(v_{i+1,j} - v_{i-1,j}) - \Delta t r v_{i,j} = 0$$

$$v_{i,j} - v_{i,j-1} + (a_i - b_i)v_{i-1,j} + (-2a_i - \Delta t r)v_{i,j} + (a_i + b_i)v_{i+1,j} = 0$$

$$v_{i,j} - v_{i,j-1} + d_i v_{i-1,j} + m_i v_{i,j} + u_i v_{i+1,j} = 0$$

$$d_i = a_i - b_i, \quad m_i = -2a_i - \Delta t r, \quad u_i = a_i + b_i$$

Matrix Formulation

d_i, m_i, u_i 를 이용하여 식을 정리하면,

$$v_{1,j} - v_{1,j-1} + d_1 v_{0,j} + m_1 v_{1,j} + u_1 v_{2,j} = 0$$

$$v_{2,j} - v_{2,j-1} + d_2 v_{1,j} + m_2 v_{2,j} + u_2 v_{3,j} = 0$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$$

$$v_{i,j} - v_{i,j-1} + d_i v_{i-1,j} + m_i v_{i,j} + u_i v_{i+1,j} = 0$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$$

$$v_{N-1,j} - v_{N-1,j-1} + d_{N-1} v_{N-2,j} + m_{N-1} v_{N-1,j} + u_{N-1} v_{N,j} = 0$$

Matrix notation

$$v_{1:N-1,j} - v_{1:N-1,j-1} + Pv_{0:N,j} = 0$$

Explicit method

Matrix notation: $v_{1:N-1,j} - v_{1:N-1,j-1} + Pv_{0:N,j} = 0$

$$P = \begin{bmatrix} d_1 & m_1 & u_1 & 0 & \dots & \dots & 0 \\ 0 & d_2 & m_2 & u_2 & & & \vdots \\ \vdots & & d_3 & m_3 & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & u_{N-2} & 0 \\ 0 & \dots & \dots & 0 & d_{N-1} & m_{N-1} & u_{N-1} \end{bmatrix} \equiv [B_{:,1} \ A \ B_{:,2}]$$
$$\Rightarrow Pv_{\Omega}^j = Av_{\Omega}^j + Bv_{\partial\Omega}^j$$
$$\Rightarrow v_{\Omega}^j - v_{\Omega}^{j-1} + Av_{\Omega}^j + Bv_{\partial\Omega}^j = 0$$

Explicit Method: $v_{\Omega}^{j-1} = (I + A)v_{\Omega}^j + Bv_{\partial\Omega}^j$

Implicit method

The derivative of V with respect to time, with a forward difference:

$$V_t|_{t=j\Delta t} \approx \frac{v_{i,j+1} - v_{i,j}}{\Delta t}$$

$$v_{\Omega}^{j+1} - v_{\Omega}^j + Av_{\Omega}^j + Bv_{\partial\Omega}^j = 0$$

$$(I - A)v_{\Omega}^j = v_{\Omega}^{j+1} + Bv_{\partial\Omega}^j$$

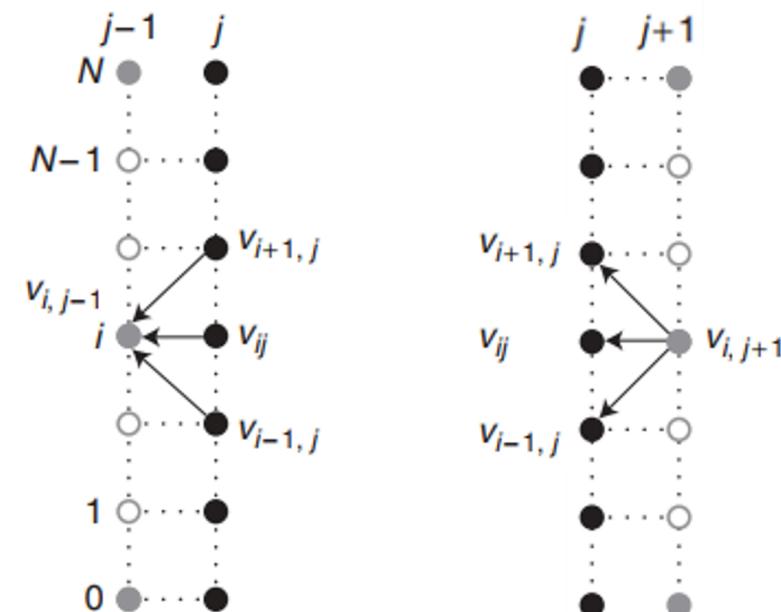


Figure 4.6 Explicit method (left) and implicit method (right).

θ -methods

Linear combination of the two previous equations:

$$\theta(v_{\Omega}^{j+1} - v_{\Omega}^j + Av_{\Omega}^j + Bv_{\partial\Omega}^j) + (1 - \theta)(v_{\Omega}^{j+1} - v_{\Omega}^j + Av_{\Omega}^{j+1} + Bv_{\partial\Omega}^{j+1}) = 0$$

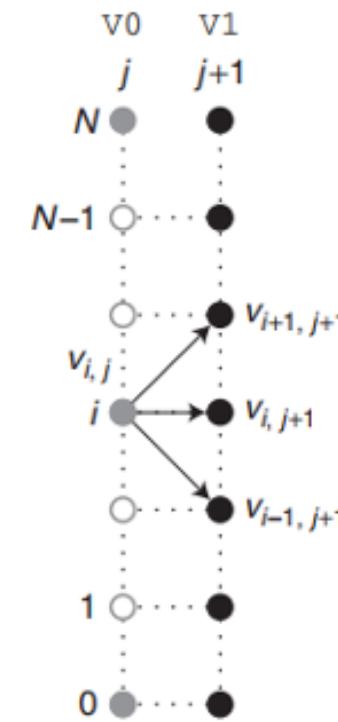
$$A_m v_{\Omega}^j = A_p v_{\Omega}^{j+1} + \theta B v_{\partial\Omega}^j + (1 - \theta) B v_{\partial\Omega}^{j+1}$$

where $A_m = I - \theta A$ and $A_p = I + (1 - \theta)A$.

$$\underbrace{A_m v_{\Omega}^j}_{V0(1:N-1)} = \underbrace{A_p v_{\Omega}^{j+1}}_{V1(1:N-1)} + \underbrace{\theta B v_{\partial\Omega}^j}_{V0([0 N])} + \underbrace{(1 - \theta) B v_{\partial\Omega}^{j+1}}_{V1([0 N])}$$

Crank-Nicolson method: $\theta = 0.5$ 인 케이스

θ	Method	A_m	A_p
0	Explicit	I	$I + A$
1	Implicit	$I - A$	I
$\frac{1}{2}$	Crank-Nicolson	$I - \frac{1}{2}A$	$I + \frac{1}{2}A$



θ -methods 구현

θ -method 구현

- 선형방정식을 연쇄적으로 풀어서 v_{Ω}^0 의 값까지 계산함
-

Algorithm 13 θ -method.

```
1: define terminal and boundary conditions  $v_{\Omega}^M$ ,  $v_{\partial\Omega}^{0:M-1}$ , and  $\theta$ 
2: compute  $A_m$ ,  $A_p$ , and  $B$ 
3: for  $j = M - 1 : -1 : 0$  do
4:   solve  $A_m v_{\Omega}^j = A_p v_{\Omega}^{j+1} + \theta B v_{\partial\Omega}^j + (1 - \theta) B v_{\partial\Omega}^{j+1}$ 
5: end for
```

Stability

FDM의 정밀도

- 시간과 공간의 step 개수에 따라 달라짐
- 선형방정식의 ill-conditioned에 의한 안정성 문제와 연관됨
- 공간 단계의 수를 늘리면 선형시스템의 크기가 증가하고, 시간 단계수는 선형시스템을 푸는 회수를 결정함

Time steps에 따른 가격 오차

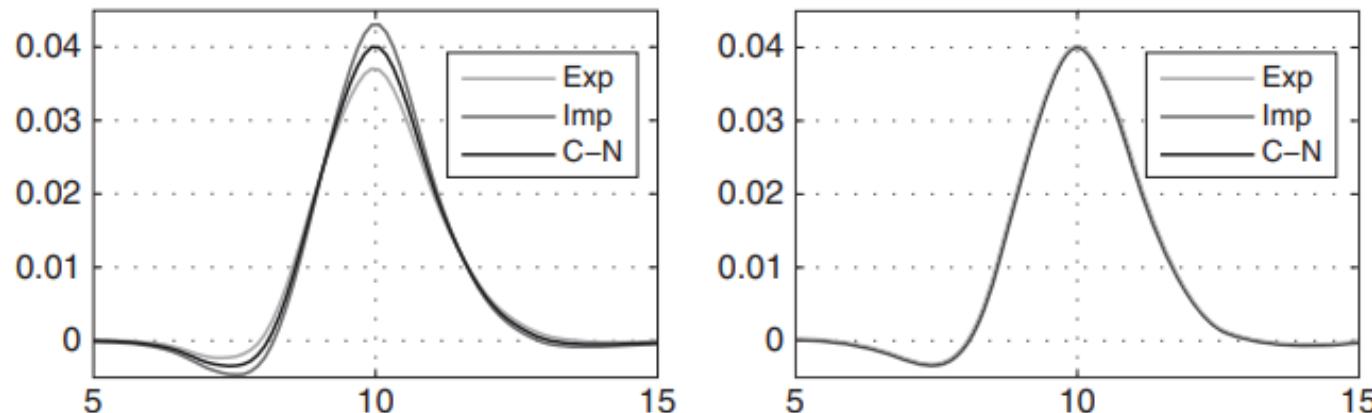


Figure 4.7 $M = 30$ (left panel) and $M = 500$ (right panel). Common parameters:
 $N = 100$, $\sigma = 0.20$, $r = 0.05$, $q = 0$, $X = S = 10$.

Stability

변동성 변화에 따른 안정성

- Explicit 방법은 낮은 변동성에서는 가장 작은 오차를 보이지만, 변동성이 높아지면 수렴하지 않고 불안정한 결과(매우 극심한 oscillation)를 보임

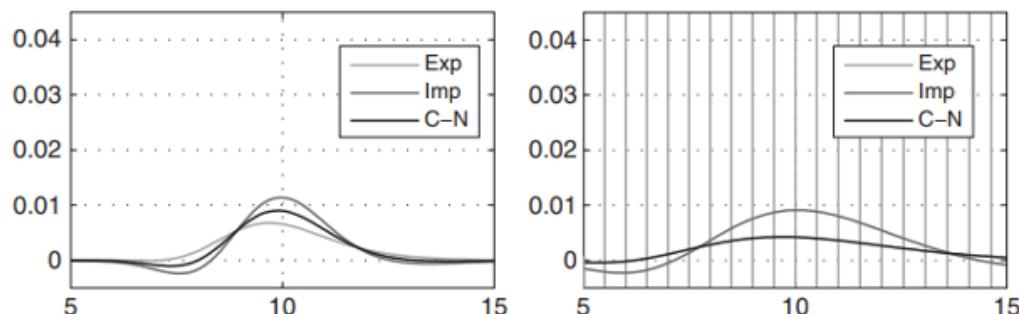


Figure 4.8 $\sigma = 0.20$ (left panel) and $\sigma = 0.40$ (right panel). Common parameters: $N = 100$, $M = 30$, $r = 0.05$, $q = 0$, $X = S = 10$.

공간 스텝 수 변화에 따른 안정성

- Explicit 방법은 공간 스텝 수가 증가하면 불안정한 결과를 보임

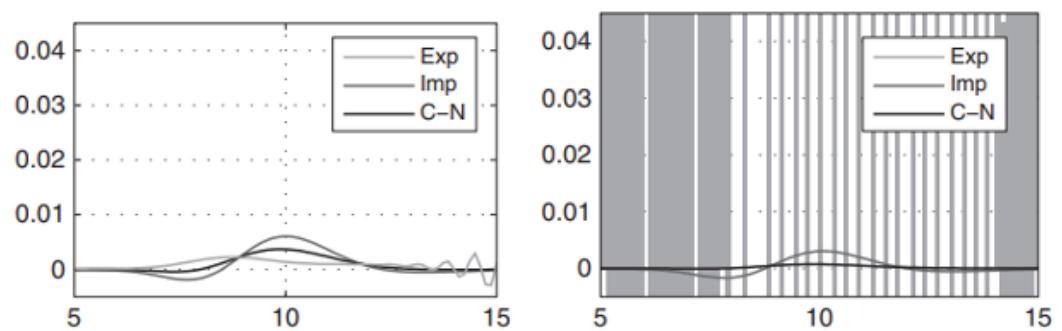


Figure 4.9 $N = 310$ (left panel) and $N = 700$ (right panel). Common parameters: $M = 30$, $\sigma = 0.20$, $r = 0.05$, $q = 0$, $X = 10$.

PDE의 변형

BS PDE

$$dS_t = (r - q)S_t dt + \sigma S_t dW_t \Rightarrow V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - q)SV_S = rV$$

- By Ito's lemma,

$$\begin{aligned} d\ln S_t &= dx_t = \left(r - q - \frac{1}{2}\sigma^2 \right) dt + \sigma dW_t \quad \text{where } x_t = \ln S_t \\ &\Rightarrow V_t + \frac{1}{2}\sigma^2 V_{xx} + \left(r - q - \frac{1}{2}\sigma^2 \right) V_x = rV \end{aligned}$$

- PDE의 계수에서 S 항이 사라지면서, 등간격 grid를 이용할 때 안정적인 계산 가능

Terminal Condition

- Call Option

$$V(x, T) = (e^x - K)^+$$

- Put Option

$$V(x, T) = (K - e^x)^+$$

American exercise

American option의 PDE:

$$\begin{cases} \mathcal{L}_{BS}(V)(V(S,t) - V(S,T)) = 0 \\ \mathcal{L}_{BS}(V) \geq 0 \quad \text{and} \quad (V(S,t) - V(S,T)) \geq 0 \end{cases} \implies \begin{cases} (A_m v_{\Omega}^j - b^j) \dot{\times} (v_{\Omega}^j - v_{\Omega}^M) = 0, \\ A_m v_{\Omega}^j \geq b^j \quad \text{and} \quad v_{\Omega}^j \geq v_{\Omega}^M \end{cases}$$

$$\text{where } \mathcal{L}_{BS}(V) \equiv V_t + \frac{1}{2} \sigma^2 S^2 V_{SS} + rSV_S - rV$$

$\dot{\times}$ 는 element by element product를 의미함

Free boundary problem

- 경계가 미리 정해지지 않은 상태에서 편미분 방정식을 풀어야 한다는 뜻
- 경계를 정하기 위한 추가 경계 조건
 - 아메리칸 옵션 문제에서는 옵션의 가치와 그 델타가 페이오프 함수와 연속적이라는 이른바 'smooth pasting condition'을 만족함
- 경계 자체를 직접 찾으려는 시도를 하지 않고, free boundary를 포함하는 기초 자산의 고정된 범위에서 문제를 해결 할 수 있음

FDM으로 American option 평가

Explicit method에서 조기 행사

- Explicit method에서는 이항모형에서 평가하는 것과 마찬가지로 V 값과 exercise value를 비교하여 큰 값을 대입 함

$$v^k = \max(v_{\text{cont}}^k, \text{exercise value})$$

- 조기 행사 경계는 옵션 가치가 페이오프와 같은 자산 값의 양쪽 사이 어딘가에 존재
 - 한쪽에서는 옵션 값이 페이오프와 동일하고, 다른 한쪽에서는 옵션 값이 페이오프보다 큼
 - 추가적인 노력 없이 모든 경계를 찾을 수 있음

Crank-Nicolson method

PSOR (projected successive overrelaxation)

- PSOR을 이용하여 반복 방법으로 선형방정식의 해를 계산

$$x_i = \max(x_i, v_i^\Omega)$$

Algorithm 14 Projected successive overrelaxation (PSOR).

```
1: give starting solution  $x \in \mathbb{R}^n$ 
2: while not converged do
3:   for  $i = 1 : n$  do
4:      $x_i = \omega(b_i^j - A_{i,1:n}x) / a_{ii} + x_i$ 
5:      $x_i = \max(x_i, v_i^\Omega)$ 
6:   end for
7: end while
8:  $v_\Omega^j = x$ 
```
