有限差分法
Finite Difference Methods

昀騰金融科技

技術長

董夢雲 博士

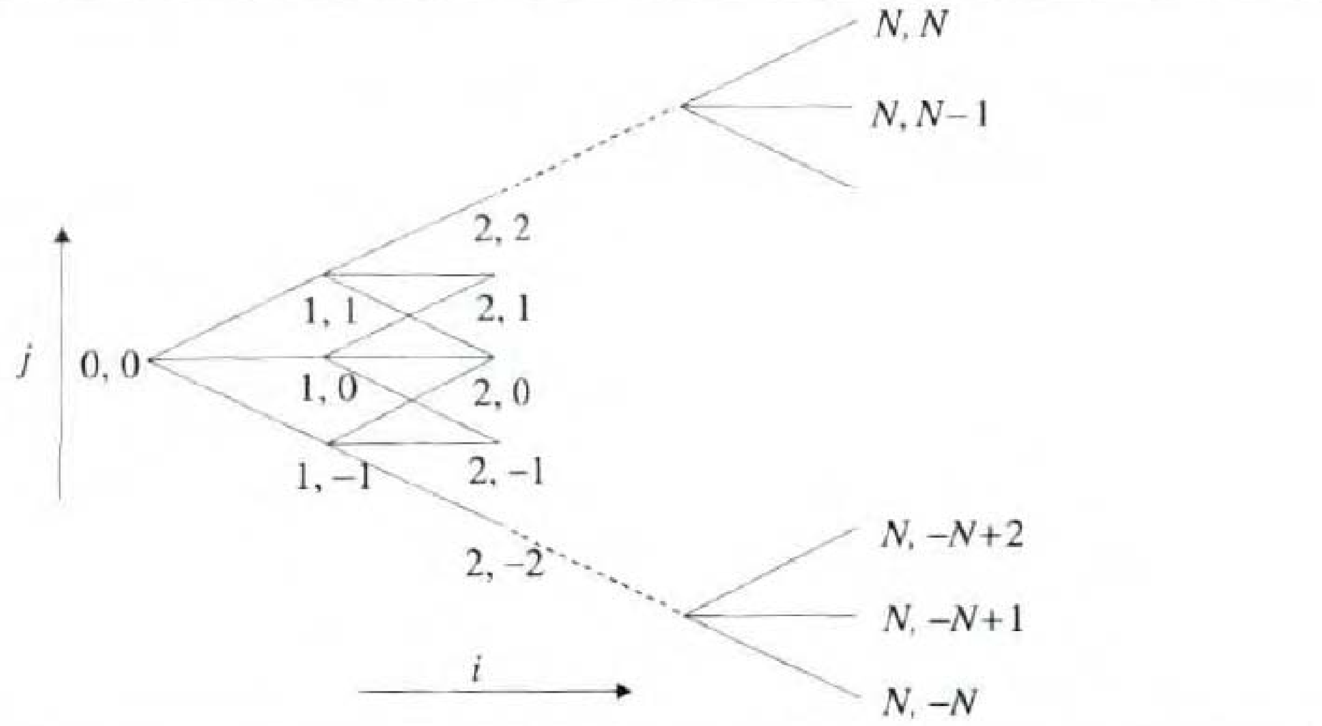
dongmy@ms5.hinet.net

目　　錄

一、由樹(Tree)到晶格(Grid)

◆ 考慮一個三元樹如下，

FIGURE 3.2 A Trinomial Tree

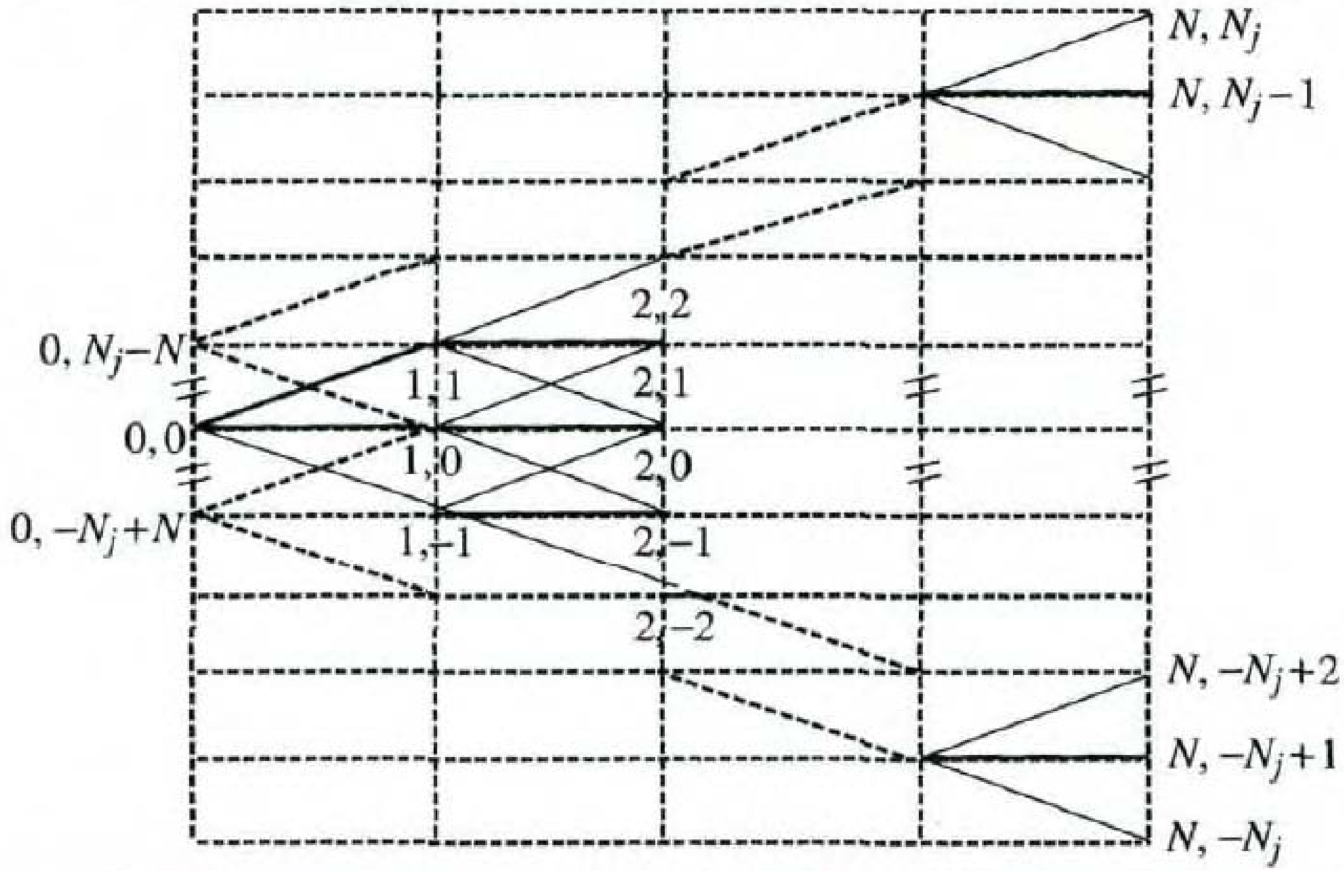


➢ 往上、下擴展後，形成有 2Nj+1 個節點的晶格體，Nj>=N，如下圖。

◆ 只有在實線上的點，選擇權之值才可求得，用折現法由後推算。

➢ 需要額外邊界條件(Boundary Conditions)，極高、極低價格時選擇權價值，才可求得其他點價值。

FIGURE 3.5 A Trinomial Tree in a Rectangular Grid



◆ 對歐式選擇權，BC 為：

$$\frac{\partial C}{\partial \mathrm{S}} = 1, \quad S >> 0$$

$$\frac{\partial C}{\partial \mathrm{S}} = 0, \quad S \approx 0$$

➢ 以差分表示，

$$\frac{C_{i,N_j} - C_{i,N_{j-1}}}{S_{i,N_j} - S_{i,N_{j-1}}} = 1, \quad S >> 0$$

$$\frac{C_{i,N_j} - C_{i,N_{j-1}}}{S_{i,N_j} - S_{i,N_{j-1}}} = 0, \quad S \approx 0$$

➢ 可以由內部向外部上、下兩方($C_{i,-Nj}$與 $C_{i,Nj}$)計算。

✓ 內部的選擇權值($C_{i,-Nj+1}$，…，$C_{i,Nj-1}$)，可由三元樹折現公式求得。

一、顯式差分法(Explicit FDM)

- 以有限差分取代偏微分的方法，簡化 PDE(Partial Differential Equation)，以求得其解。
  - 回到 Black-Scholes PDE 如下式。

$$-\frac{\partial C}{\partial t}=\frac{1}{2}S^2\sigma^2\frac{\partial^2 C}{\partial S^2}+(r-y)S\frac{\partial C}{\partial S}-rC \quad \text{..........(1)}$$

  - 不同的選擇權有其各自的邊界條件(Boundary Conditions)。

- 以 x = ln(S)表示，(1)式改寫如下，

$$-\frac{\partial C}{\partial t}=\frac{1}{2}\sigma^2\frac{\partial^2 C}{\partial x^2}+\upsilon\frac{\partial C}{\partial x}-rC \quad \text{..........(2)}$$

  - (2)式為一常數係數的 PDE，與 x、t 無關，容易求解。

◆ 使用前向差分的方法來處理 t 的微分，使用中央差分的方法來處理 x 的微分。

- 對於使用中央差分的方法來處理 x 的微分，取值於 t = i+1。可以得到下式，

$$-\frac{\partial C}{\partial t}=\frac{\mathrm{C}_{i+1,j}-\mathrm{C}_{i,j}}{\Delta t}$$

$$\frac{\partial C}{\partial x}=\frac{C_{i+1,j+1}-C_{i+1,j-1}}{2\Delta x}\ ,\quad \frac{\partial^2 C}{\partial x^2}=\frac{C_{i+1,j+1}-2C_{i+1,j}+C_{i+1,j-1}}{\Delta x^2}$$

- (2)式可以改寫成下式。

$$-\frac{C_{i+1,j}-C_{i,j}}{\Delta t}=\frac{1}{2}\sigma^2\frac{C_{i+1,j+1}-2C_{i+1,j}+C_{i+1,j-1}}{\Delta x^2}+\upsilon\frac{C_{i+1,j+1}-C_{i+1,j-1}}{2\Delta x}-rC_{i+1,j} \qquad \text{.........................(3)}$$

- 進一步改寫成下式。

$$C_{i,j}=p_u C_{i+1,j+1}+p_m C_{i+1,j}+p_d C_{i+1,j-1} \qquad \text{..(4)}$$

$$p_u=\Delta t\left(\frac{\sigma^2}{2\Delta x^2}+\frac{\upsilon}{2\Delta x}\right)\ ,\quad p_m=1-\Delta t\frac{\sigma^2}{\Delta x^2}-r\Delta t\ ,\quad p_u=\Delta t\left(\frac{\sigma^2}{2\Delta x^2}-\frac{\upsilon}{2\Delta x}\right)$$

FIGURE 3.6 The Explicit Finite Difference Method

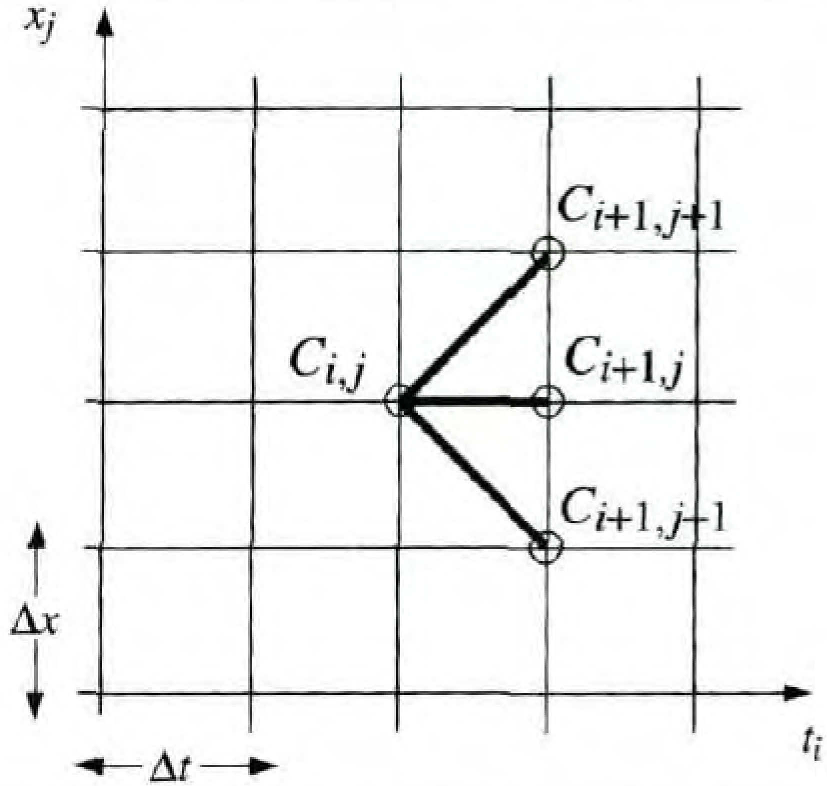


FIGURE 3.7 Pseudo-code for a European Call Option by Explicit Finite Difference Method

```
initialise_parameters { K, T, S, sig, r, div, N, Nj, dx }

{ precompute constants }

dt = T/N
nu = r - div - 0.5 * sig^2
edx = exp(dx)
pu = 0.5*dt*( (sig/dx)^2 + nu/dx )
pm = 1.0 - dt*(sig/dx)^2 - r*dt
pd = 0.5*dt*( (sig/dx)^2 - nu/dx )

{ initialise asset prices at maturity }

St[-Nj] = S*exp(-Nj*dx)
for j = -Nj+1 to Nj do St[j] = St[j-1]*edx

{ initialise option values at maturity }

for j = -Nj to Nj do C[N,j] = max( 0 , St[j] - K )

{ step back through lattice }

for i = N-1 downto 0 do

  for j = -Nj+1 to Nj-1 do
    C[i,j] =
      pu*C[i+1,j+1] + pm*C[i+1,j] + pd*C[i+1,j-1]

  { boundary conditions }

  C[i,-Nj] = C[i,-Nj+1]

  C[i,Nj] = C[i,Nj-1] + (St[Nj]-St[Nj-1])

next i

European_call = C[0,0]
```

◆ 範例：

FIGURE 3.8 Numerical Example for a European Call Option by Explicit Finite Difference Method

K	T	S	sig	r	div	N	Nj	dx
100	1	100	0.2	0.06	0.03	3	3	0.2

dt	nu	edx	pu	pm	pd
0.3333	0.0100	1.2214	0.1750	0.6467	0.1583

	i	0	1	2	3
j	St, t	0	0.3333	0.6667	1
3	182.21	83.9151	83.2738	82.7267	82.2119
2	149.18	50.8857	50.2444	49.6973	49.1825
1	122.14	25.4319	24.1349	22.9243	22.1403
0	100.00	8.5455	6.5173	3.8745	0.0000
−1	81.87	1.5790	0.6780	0.0000	0.0000
−2	67.03	0.1187	0.0000	0.0000	0.0000
−3	54.88	0.1187	0.0000	0.0000	0.0000

FIGURE 3.9 Pseudo-code for an American Put Option by Explicit Finite Difference Method

```
initialise_parameters { K, T, S, sig, r, div, N, Nj, dx }

{ precompute constants }

dt = T/N
nu = r - div - 0.5 * sig^2
edx = exp (dx)
pu = 0.5*dt*( (sig/dx)^2 + nu/dx )
pm = 1.0 - dt*(sig/dx)^2 - r*dt
pd = 0.5*dt*( (sig/dx)^2 - nu/dx )

{ initialise asset prices at maturity }

St[-Nj] = S*exp(-Nj*dx)
for j = -Nj+1 to Nj do St[j] = St[j-1]*edx

{ initialise option values at maturity }

for j = -Nj to Nj do C[0,j] = max( 0 , K - St[j] )

{ step back through lattice }

for i = N-1 downto 0 do

  for j = -Nj+1 to Nj-1 do
    C[1,j] = pu*C[0,j+1] + pm*C[0,j] + pd*C[0,j-1]

  { boundary conditions }

  C[1,-Nj] = C[1,-Nj+1] + (St[-Nj+1]-St[-Nj])

  C[1,Nj] = C[1,Nj-1]

  { apply early exercise condition }

  for j = -Nj to Nj do
    C[0,j] = max( C[1,j] , K - St[j] )

next i

American_put = C[0,0]
```

◆ 範例：

FIGURE 3.10 Numerical Example for an American Put Option by Explicit Finite Difference Method

K	T	S	sig	r	div	N	Nj	dx
100	1	100	0.2	0.06	0.03	3	3	0.2

dt	nu	edx	pu	pm	pd
0.3333	0.0100	1.2214	0.1750	0.6467	0.1583

	i	0	1	2	3
j	St, t	0	0.3333	0.6667	1
3	182.21	0.0720 0.0720	0.0000 0.0000	0.0000 0.0000	0.0000
2	149.18	0.0720 0.0720	0.0000 0.0000	0.0000 0.0000	0.0000
1	122.14	1.0422 1.0422	0.4544 0.4544	0.0000 0.0000	0.0000
0	100.00	6.0058 6.0058	4.7261 4.7261	2.8701 2.8701	0.0000
−1	81.87	17.7691 18.1269	17.4443 18.1269	16.9420 18.1269	18.1269
−2	67.03	31.6353 32.9680	31.6353 32.9680	31.6353 32.9680	32.9680
−3	54.88	43.7862 45.1188	43.7862 45.1188	43.7862 45.1188	45.1188

*discounted expectation or boundary condition

$\boxed{\begin{matrix} * \\ C \end{matrix}}$

◆ 穩定度的要求，p_u 、 p_m 、 p_d ，皆大於零，

$$\Delta x \geq \sigma\sqrt{3\Delta t}$$

➢ 到期日通常考慮上、下 3 個標準差的資產價格，$\sigma = 0.25$ ，$T = 1.0$ ，令價格為 100(2Nj+1=100)，

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

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

✓ 每一年要 833 步。

➢ 以 n_{SD} 取代 6，可得

$$\mathrm{N} = \frac{\mathrm{T}}{\Delta t} \geq 3\left(\frac{2N_j + 1}{n_{SD}}\right)^2$$

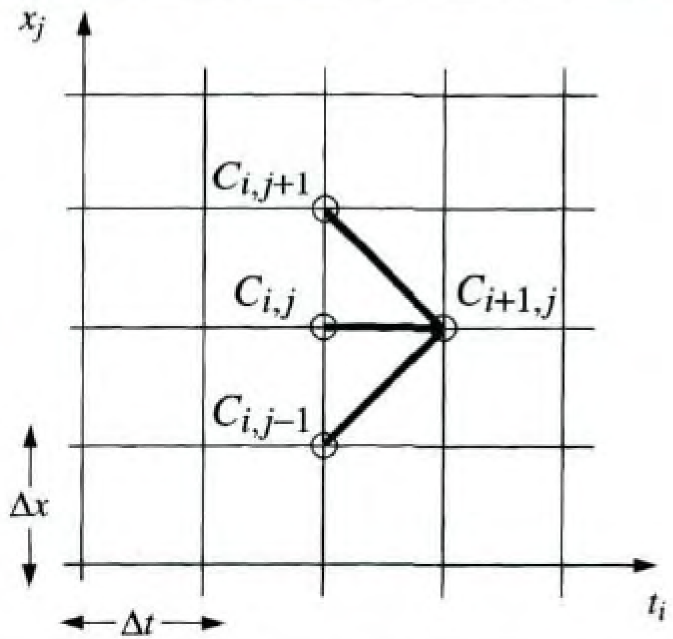
➢ 通常要求晶格上，資產價格數目對標準差數目之比大於 15。

$$\left(\frac{2N_j + 1}{n_{SD}}\right) \geq 15$$

✓ 每一年要 675 步。

二、隱式差分法(Implicit FDM)

FIGURE 3.12 The Implicit Finite Difference Method



◆ 在 Black-Scholes PDE，對於使用中央差分的方法來處理 x 的微分，取值於 t=i 而非 i+1。可以得到下式，

$$-\frac{C_{i+1,j}-C_{i,j}}{\Delta t}=\frac{1}{2}\sigma^2\frac{C_{i,j+1}-2C_{i,j}+C_{i,j-1}}{\Delta x^2}+\upsilon\frac{C_{i,j+1}-C_{i,j-1}}{2\Delta x}-rC_{i,j} \qquad \text{..............................(5)}$$

➢ 改寫成下式。

$$p_uC_{i,j+1}+p_mC_{i,j}+p_dC_{i,j-1}=C_{i+1,j} \qquad \text{..............................(6)}$$

$$p_u=-\frac{1}{2}\Delta t\left(\frac{\sigma^2}{\Delta x^2}+\frac{\upsilon}{\Delta x}\right), \quad p_m=1+\Delta t\frac{\sigma^2}{\Delta x^2}+r\Delta t, \quad p_u=-\frac{1}{2}\Delta t\left(\frac{\sigma^2}{\Delta x^2}-\frac{\upsilon}{\Delta x}\right)$$

➢ (6)式無法直接求得。需配合邊界條件，

$$C_{i,\mathrm{N}_j}-C_{i,N_{j-1}}=\lambda_U$$

$$C_{i,-\mathrm{N}_j+1}-C_{i,-N_j}=\lambda_L$$

✓ $2\mathrm{N}_j+1$ 個線性聯立方程式，可求得 i 時點 $2\mathrm{N}_j+1$ 個選擇權價值。

◆ 對於 Vanilla Call，邊界條件為，

$$C_{i,\mathrm{N}_j} - C_{i,N_{j-1}} = \lambda_U = \mathrm{S}_{i,\mathrm{N}_j} - \mathrm{S}_{i,N_{j-1}}$$

$$C_{i,-\mathrm{N}_j+1} - C_{i,-N_j} = \lambda_L = 0$$

➢ 線性方程組可表式為，

$$\begin{bmatrix} 1 & -1 & 0 & \ldots & \ldots & \ldots & 0 \\ p_u & p_m & p_d & 0 & \ldots & \ldots & 0 \\ 0 & p_u & p_m & p_d & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & p_u & p_m & p_d & 0 \\ 0 & \ldots & \ldots & 0 & p_u & p_m & p_d \\ 0 & \ldots & \ldots & \ldots & 0 & 1 & -1 \end{bmatrix} \bullet \begin{bmatrix} C_{i,\mathrm{N}_j} \\ C_{i,\mathrm{N}_j-1} \\ C_{i,\mathrm{N}_j-2} \\ \ldots \\ C_{i,-\mathrm{N}_j+2} \\ C_{i,-\mathrm{N}_j+1} \\ C_{i,-\mathrm{N}_j} \end{bmatrix} = \begin{bmatrix} \lambda_U \\ C_{i+1,\mathrm{N}_j-1} \\ C_{i+1,\mathrm{N}_j-2} \\ \ldots \\ C_{i+1,-\mathrm{N}_j+2} \\ C_{i+1,-\mathrm{N}_j+1} \\ \lambda_L \end{bmatrix}$$

➢ 由下往上，逐步遞歸求解，或反矩陣求解。

FIGURE 3.13 Pseudo-code for an American Put Option by Implicit Finite Difference Method

```
initialise_parameters { K, T, S, sig, r, div, N, Nj, dx }

{ precompute constants }

dt = T/N
nu = r - div - 0.5 * sig^2
edx = exp(dx)
pu = -0.5*dt*( (sig/dx)^2 + nu/dx )
pm = 1.0 + dt*(sig/dx)^2 + r*dt
pd = -0.5*dt*( (sig/dx)^2 - nu/dx )

{ initialise asset prices at maturity }

St[-Nj] = S*exp(-Nj*dx)
for j = -Nj+1 to Nj do St[j] = St[j-1]*edx

{ initialise option values at maturity }

for j = -Nj to Nj do C[0,j] = max( 0 , K - St[j] )

{ compute derivative boundary condition }

lambda_L = -1 * ( St[-Nj+1] - St[-Nj] )
lambda_U = 0.0

{ step back through lattice }

for i = N-1 downto 0 do

  solve_implicit_tridiagonal_system( C, pu, pm, pd,
    lambda_L, lambda_U )

  { apply early exercise condition }

  for j = -Nj to Nj do
    C[0,j] = max( C[1,j] , K - St[j] )

next i

American_put = C[0,0]

{---------------------------------------------------------------------------}
```

(continues)

FIGURE 3.13 *(continued)*

```
subroutine solve_implicit_tridiagonal_system( C, pu, pm, pd,
  lambda_L, lambda_U )

{ substitute boundary condition at j = -Nj into j = -Nj+1 }

pmp[-Nj+1] = pm + pd
pp[-Nj+1] = C[0,-Nj+1] + pd*lambda_L

{ eliminate upper diagonal }

for j = -Nj+2 to Nj-1 do
  pmp[j] = pm - pu*pd/pmp[j-1]
  pp[j] = C[0,j] - pp[j-1]*pd/pmp[j-1]
next j

{ use boundary condition at j = Nj and equation at j = Nj-1 }

C[1,Nj] = (pp[Nj-1] + pmp[Nj-1]*lambda_U)/(pu + pmp[Nj-1])
C[1,Nj-1] = C[1,Nj] - lambda_U

{ back-substitution }

for j = Nj-2 downto -Nj do
  C[1,j] = ( pp[j] - pu*C[1,j+1] )/pmp[j]
next j

return
```

◆ 範例：

FIGURE 3.14 Numerical Example for an American Put Option by Implicit Finite Difference Method

K	T	S	sig	r	div	N	Nj	dx
100	1	100	0.2	0.06	0.03	3	3	0.2

dt	nu	edx	pu	pm	pd
0.3333	0.0100	1.2214	−0.1750	1.3533	−0.1583

	i	0		1		2		3
j	St, t	0		0.3333		0.6667		1
3	182.21	0.2386 0.2386		0.1120 0.1120		0.0330 0.0330		0.0000
2	149.18	0.2386 0.2386	1.3325 0.2762	0.1120 0.1120	1.3325 0.1296	0.0330 0.0330	1.3325 0.0382	0.0000
1	122.14	1.0684 1.0684	1.3325 1.3819	0.6248 0.6248	1.3325 0.8129	0.2458 0.2458	1.3325 0.3218	0.0000
0	100.00	4.9221 4.9221	1.3325 6.3718	3.6637 3.6637	1.3325 4.7726	2.0646 2.0646	1.3325 2.7060	0.0000
−1	81.87	17.7509 18.1269	1.3301 22.7500	17.5854 18.1269	1.3301 22.7500	17.3750 18.1269	1.3301 22.7500	18.1269
−2	67.03	31.7977 32.9680	1.1950 34.8919	31.7735 32.9680	1.1950 34.8919	31.7427 32.9680	1.1950 34.8919	32.9680
−3	54.88	43.9486 45.1188		43.9243 45.1188		43.8935 45.1188		45.1188

*solution of tridiagonal system

pmp pp	* C

三、Crank-Nicolson FDM

- 又稱為完全中心法(Full Centered Method)，是隱式 FDM 法的修正版。時間與空間的微分皆取在 t=i+1/2 上。

FIGURE 3.15 The Crank–Nicolson Finite Difference Method

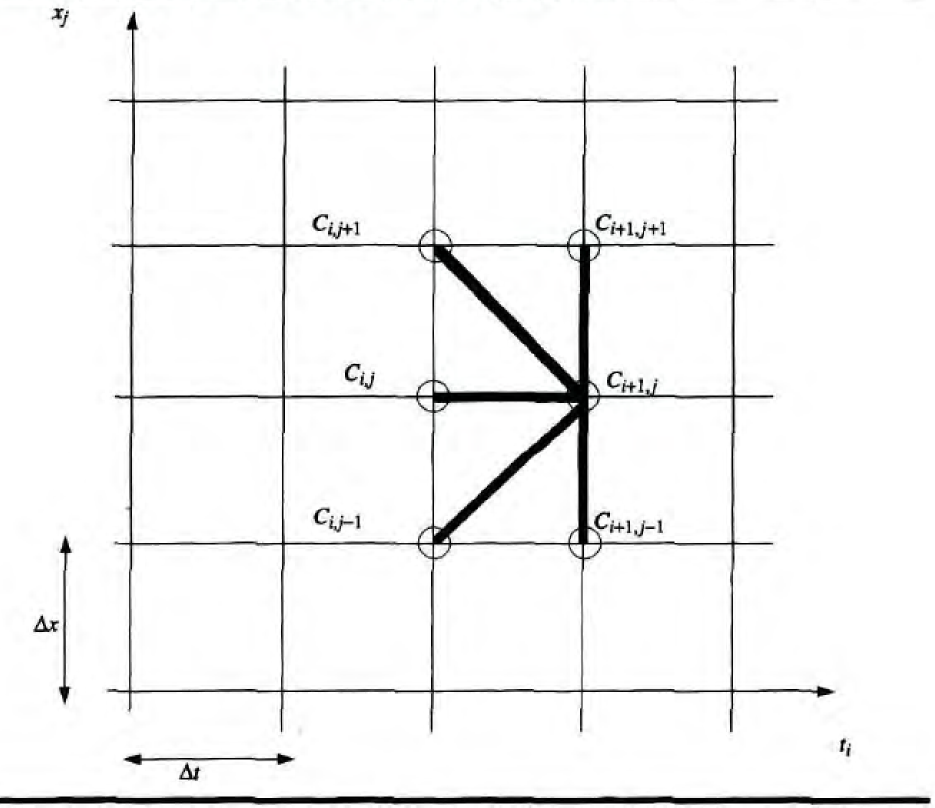


FIGURE 3.16 Pseudo-code for an American Put Option by the Crank–Nicolson Finite Difference Method

```
initialise_parameters { K, T, S, sig, r, div, N, Nj, dx }

{ precompute constants }

dt = T/N
nu = r - div - 0.5 * sig^2
edx = exp(dx)
pu = -0.25*dt*( (sig/dx)^2 + nu/dx )
pm = 1.0 + 0.5*dt*(sig/dx)^2 + 0.5*r*dt
pd = -0.25*dt*( (sig/dx)^2 - nu/dx )

{ initialise asset prices at maturity }

St[-Nj] = S*exp(-Nj*dx)
for j = -Nj+1 to Nj do St[j] = St[j-1]*edx

{ initialise option values at maturity }

for j = -Nj to Nj do C[0,j] = max( 0 , K - St[j] )

{ compute derivative boundary condition }

lambda_L = -1 * ( St[-Nj+1] - St[-Nj] )
lambda_U = 0.0

{ step back through lattice }

for i = N-1 downto 0 do

  solve_Crank_Nicolson_tridiagonal_system( C, pu, pm, pd,
    lambda_L, lambda_U )

  { apply early exercise condition }

  for j = -Nj to Nj do
```

FIGURE 3.16 (continued)

```
    C[0,j] = max( C[1,j] , K - St[j] )
next i

American_put = C[0,0]

{-----------------------------------------------------------------------------}

subroutine solve_Crank_Nicolson_tridiagonal_system( C,
  pu, pm, pd, lambda_L, lambda_U )

{ substitute boundary condition at j = -Nj into j = -Nj+1 }

pmp[-Nj+1] = pm + pd
pp[-Nj+1] = -pu*C[0,-Nj+2]-(pm-2)*C[0,-Nj+1]-pd*C[0,-Nj]
              + pd*lambda_L

{ eliminate upper diagonal }

for j = -Nj+2 to Nj-1 do
  pmp[j] = pm - pu*pd/pmp[j-1]
  pp[j] = -pu*C[0,j+1]-(pm-2)*C[0,j]-pd*C[0,j-1]
          - pp[j-1]*pd/pmp[j-1]
next j

{ use boundary condition at j = Nj and equation at j = Nj-1 }

C[1,Nj] = (pp[Nj-1] + pmp[Nj-1]*lambda_U)/(pu + pmp[Nj-1])
C[1,Nj-1] = C[1,Nj] - lambda_U

{ back-substitution }

for j = Nj-2 downto -Nj do
  C[1,j] = ( pp[j] - pu*C[1,j+1] )/pmp[j]
next j

return
```

◆ 範例：

FIGURE 3.17 Numerical Example for an American Put Option by the Crank–Nicolson Finite Difference Method

K	T	S	sig	r	div	N	Nj	dx
100	1	100	0.2	0.06	0.03	3	3	0.2

dt	nu	edx	pu	pm	pd
0.3333	0.0100	1.2214	−0.0875	1.1767	−0.0792

	i	0	1	2	3
j	St/t	0	0.3333	0.6667	1
3	182.21	0.1686 0.1686	0.0620 0.0620	0.0117 0.0117	0.0000
2	149.18	0.1686 0.1686	0.0620 0.0620	0.0117 0.0117	0.0000
1	122.14	1.0488 1.0488	0.5568 0.5568	0.1616 0.1616	0.0000
0	100.00	5.4184 5.4184	4.1237 4.1237	2.3896 2.3896	0.0000
−1	81.87	17.7458 18.1269	17.5193 18.1269	17.2110 18.1269	18.1269
−2	67.03	31.7233 32.9680	31.7053 32.9680	31.6807 32.9680	32.9680
−3	54.88	43.8742 45.1188	43.8561 45.1188	43.8315 45.1188	45.1188

*solution of tridiagonal system

*

C