從標的物價格隨機過程到數值評價模型

Financial Engineering and Computations

Dai, Tian-Shyr

授課大綱

- Wiener process (Brownian motion)定義及特性
- Ito's process
- Ito's integral
- Ito's lemma
- 雙標的物的商品價格隨機過程
- 數值評價模型建構
  - CRR二元樹評價 (C++財務程式設計第七章)
  - 組合數學處理二元樹的評價 (C++財務程式設計第八章)
  - 蒙地卡羅法 (下一章)

Wiener Process
(Brownian motion)

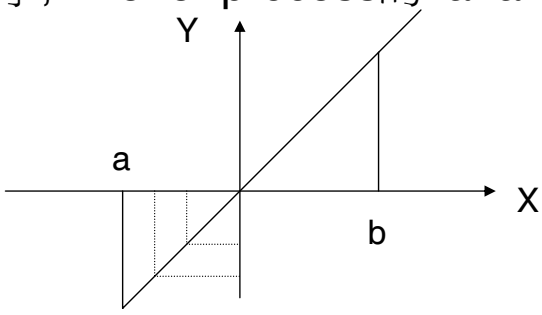
- 爲一連續時間的隨機過程 (用B(t)表示)
  - 上一章提到的丟銅板的過程爲離散隨機過程
- Wiener Process包含下列性質:
  - B(t)-B(s)~N(0,t-s)
    - N(u,v)代表常態隨機變數,平均爲u,變異數爲v
  - B(t)是一個 martingale process
    - $E(B(t)\,|\,F_s) = E(B(t)-B(s)\,|\,F_s) + E(B(s)\,|\,F_s) = B(s)$
  - 假定 $u \le s \le t$, 則B(t)-B(s)和B(u)獨立
  - B(t)爲一連續函數

Wiener Process的重要特性

- Markov process
  - 當隨機過程現在的狀態已知,則該過程未來行爲和該過程的過去行爲彼此獨立
  - 假定X(t)是 Markov process
    - $P(X(t+s) \le y\,|\,F_t) = P(X(t+s) \le y\,|\,X(t))$
  - 假定一個價格(或利率,指數)隨機過程爲一 Markov process:
    - 現在價格完全反映市場已知的訊息
- Wiener process是 Markov process
  - Black-Scholes 模型中的標的物價格函數也是 Markov process.

Wiener Process的重要特性

- 每個點都不可微分
  - $\frac{dB(t)}{dt}$ 不存在
- Infinite variation:
  - variation定義:假定在(a,b)中,切n個點:x1,x2,…xn
    - $\lim_{\|P\|\to 0}\sum_{i=1}^{n}\left|f(x_i)-f(x_{i-1})\right|$
    - finite variation的例子: y=2x, variation=2(b-a)
    - 無論(a,b)多小, Wiener process的variation都是無限大



Quadratic Variation的定義和性質

- quadratic variation: $\lim_{\|p\|\to 0}\sum_{i=1}^{n}\left|f(x_i)-f(x_{i-1})\right|^2$
  - Finite variation的連續函數, quadratic variation=0

$$\lim_{\|P\|\to 0}\sum_{i=1}^{n}\left|f(x_i)-f(x_{i-1})\right|^2 \le \lim_{\|P\|\to 0}\max_{1\le i\le n}\left|f(x_i)-f(x_{i-1})\right|\sum_{i=1}^{n}\left|f(x_i)-f(x_{i-1})\right|=0$$

$$\lim_{\|P\|\to 0}\left|f(x_i)-f(x_{i-1})\right|^2=0$$

$$\Delta t\to 0,(f(t+\Delta t)-f(t))^2=0\Rightarrow(df(t))^2=0$$

Wiener Process的Quadratic Variation

- Wiener process has infinite variation
  $B(t+\Delta t)-B(t)\sim N(0,\Delta t)$
  – Wiener process 在(a,b)Quadratic variation=b-a

$$\Delta t\to 0, B(t+\Delta t)-B(t)\equiv \varepsilon\sqrt{\Delta t}\quad (\varepsilon\sim N(0,1))$$
$$\mathrm{E}\left[(B(t+\Delta t)-B(t))^2\right]=E\left(\varepsilon^2\Delta t\right)=\Delta t$$
$$\mathrm{Var}\left[(B(t+\Delta t)-B(t))^2\right]=\mathrm{E}\left[\left(\varepsilon^2\Delta t-\Delta t\right)^2\right]=\Delta t^2\mathrm{E}\left[\left(\varepsilon^2-1\right)^2\right]\to 0$$
$$\Rightarrow (dB(t))^2=dt$$

- 上述性質可幫助推導Ito's lemma

Ito's Process

- 用Wiener process和其他函數,來構成較複雜的隨機過程
- 構成的隨機過程常用微分符號表示
  – Wiener process: $dB(t)$
  – 令 $dX(t)=adt+bdB(t)$
    - E(X(t)-X(s))=a(t-s) , Var(X(t)-X(s))= $b^2(t-s)$
    - a: drift term (代表趨勢) b: 代表波動大小
  – 令 $dX(t)=\mu X(t)dt+\sigma X(t)dB(t)$
    - Black-Scholes Formula 假定的標的物價格的隨機過程
    - μ 代表報酬率, σ 代表價格波動

課堂練習: martingale and Markov process

- 假定S(t) 是 martingale process
  - 則 $u \le t$ $E(S(t)\,|\,\mathrm{F_u}) = S(u)$
  - Let dX(t)=bdS(t), 說明 X(t)是 martingale process.
- 假定S(t)是 Markov process
  - 則 $P(S(t+s) \le y \,|\, F_t) = P(S(t+s) \le y \,|\, S(t))$
  - Let dX(t)=bdS(t),S(0)=X(0)=0, 說明 X(t)是 Markov process.

Ito's Integral

- 考慮 Ito's process的積分
- 令simple process: $Y(t) = C_0 I_0(t) + \sum_{i=0}^{n-1} C_i I_{(t_i, t_{i+1}]}(t)$
- 考慮Y(t)的積分

$$\int Y(t)dB(t) = \sum_{i=0}^{n-1} C_i (B(t_{i+1}) - B(t_i))$$

  - 爲一常態的隨機變數:
  - Mean: $E\left(\int Y(t)dB(t)\right) = 0$

Ito's Integral

– Variance:

$$\mathrm{Var}\left(\int Y(t)dB(t)\right) = \mathrm{E}\left(\sum_{i=0}^{n-1} C_i(B(t_{i+1}) - B(t_i))\right)^2$$

$$= \mathrm{E}\left(\sum_{i=0}^{n-1} C_i(B(t_{i+1}) - B(t_i)) \times \sum_{i=0}^{n-1} C_i(B(t_{i+1}) - B(t_i))\right)$$

$$= \sum_{i=0}^{n-1}\sum_{\mathrm{j}=0}^{n-1} \mathrm{E}\left(C_i C_j (B(t_{i+1}) - B(t_i))(B(t_{j+1}) - B(t_j))\right)$$

$$= \sum_{i=0}^{n-1} C_i^{\,2}(t_{i+1} - t_i) = \int \left(Y(t)\right)^2 dt$$

Ito's Integral

- 令 $dX(t) = adt + bdB(t)$
- 考慮 Y(t)dX(t)的積分
  - $\int Y(t)dX(t) = \int aY(t)dt + \int bY(t)dB(t)$

    $\mathrm{E}\int Y(t)dX(t) = \int aY(t)dt$

    $\mathrm{Var}\int Y(t)dX(t) = \int \left(bY(t)\right)^2 dt$

  – 可用來解釋投資組合的價值變化

    $\int Y(t)dX(t) = \sum Y(t_i)(X(t_i) - X(t_{i-1}))$

  – Y(t): 投資組合，X(t):標的物價格

用泰勒展開式推Ito's lemma

- 假定f(x) 及其高階導函數都可微
- $f(x)-f(x_0)$可用下列式子逼近

$$f(x)-f(x_0)=f^{'}(x_0)(x-x_0)+\frac{1}{2}f^{''}(x_0)(x-x_0)^2$$

$$+\ldots+\frac{1}{n!}f^{(n)}(x_0)(x-x_0)^n+\text{Error Term}$$

- 假定X(t)爲一隨機過程,仿照上式

$$f(X(t))-f(X(0))=f^{'}(X(0))(X(t)-X(0))+\frac{1}{2}f^{''}(X(0))(X(t)-X(0))^2$$

$$+\ldots+\frac{1}{n!}f^{(n)}(X(0))(X(t)-X(0))^n+\text{Error Term}$$

Ito's Lemma

- 考慮$t\to 0$ $X(t)-X(0)=dX(t)$
  - $df(X(t))=f^{'}(X(t))dX(t)+\frac{1}{2}f^{''}(X(t))(dX(t))^2$
- 假定X(t) is finite variation(見<u>前面描述</u>)
  - $df(X(t))=f^{'}(X(t))dX(t)$ (連鎖規則)
- 假定X(t)爲Wiener process B(t)(見<u>前面描述</u>)
  - $df(B(t))=f^{'}(B(t))dB(t)+\frac{1}{2}f^{''}(B(t))(dB(t))^2$

$$=f^{'}(B(t))dB(t)+\frac{1}{2}f^{''}(B(t))dt$$

Ito's Lemma

- 假定X(t)是 Ito's process
  - 令 $dX(t) \equiv adt + bdB(t)$
  - 則 $\left(dX(t)\right)^2 = \left(adt + bdB(t)\right)^2 = b^2 dt$

$$df(X(t)) = f'(X(t))dX(t) + \frac{1}{2} f''(X(t))(dX(t))^2$$

$$= f'(X(t))(adt + bdB(t)) + \frac{1}{2} f''(X(t))b^2 dt$$

$$= \left[af'(X(t)) + \frac{1}{2} b^2 f''(X(t)) \right] dt + \left[bf'(X(t)) \right] dB(t)$$

課堂練習: 對數常態價格過程

- 令 $f(x) = e^x$ $\quad dX(t) = (r - \frac{1}{2}\sigma^2)dt + \sigma dB(t)$
- 求 df(X(t))
- 驗證 $df(X(t)) = rf(X(t))dt + \sigma f(X(t))dB(t)$

- 令X(0)=0, $X(t) = (r - \frac{1}{2}\sigma^2)t + \sigma B(t)$
- Black-Scholes formula所假定的標的物價格函數 S(t),在風險中立機率下,可寫成

$$S(t) = S(0)e^{(r - \frac{1}{2}\sigma^2)t + \sigma B(t)}$$

Ito's Lemma的延伸

- f(t,X(t))的Ito's lemma 可表示如下:

$$df(t,X(t)) = \frac{\partial f}{\partial t}dt + \frac{\partial f(X(t))}{\partial X(t)}dX(t) + \frac{1}{2}\frac{\partial^2 f(X(t))}{\partial X(t)^2}(dX(t))^2$$

- 考慮標的物的遠期價格 $f(S,t) \equiv Se^{r(T-t)}$
  - 標的物的隨機過程: $dS(t) = \mu S(t)dt + \sigma S(t)dB(t)$

$$\begin{aligned} df(S,t) &= -rSe^{r(T-t)}dt + e^{r(T-t)}dS(t) + \frac{1}{2}\times 0(dS(t))^2 \\ &= -rf(S,t)dt + e^{r(T-t)}(\mu S(t)dt + \sigma S(t)dB(t)) \\ &= (\mu - r)f(S,t)dt + \sigma f(S,t)dB(t) \end{aligned}$$

課堂練習:遠期價格的隨機過程

- 風險中立機率下,標的物的報酬爲無風險利率 $dS(t) = \mu S(t)dt + \sigma S(t)dB(t), \rightarrow \mu = r$
  - 請描述風險中立機率下的遠期價格的隨機過程
  - 假定遠期價格的隨機過程可表成 $f(t) = f(0)e^{X(t)}$
    請描述 dX(t)

兩個隨機過程下的Ito's Process

- 考慮市場上有兩個資產,隨機過程如下:
  - $$dX_1 = a_1(X_1,t)dt + b_1(X_1,t)dB_1(t)$$
    $$dX_2 = a_2(X_2,t)dt + b_2(X_2,t)dB_2(t)$$
  - $B_1(t)$ $B_2(t)$爲兩個不同的 Wiener process
- 給定衍生性商品D,其標的物爲上述兩個資產,D的價格波動可經Ito's Process描述
  - $dB_1(t)$ $dB_2(t)$的計算和$B_1(t)$ $B_2(t)$ 的相關係數ρ有關: $dB_1(t)dB_2(t) = \rho dt$
  - Remark: $\rho_{xy} = \dfrac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y}$

兩個隨機過程下的Ito's Process

$$\begin{aligned}
df(t, X_1(t), X_2(t)) &= \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial X_1(t)}dX_1(t) + \frac{1}{2}\frac{\partial^2 f}{\partial X_1(t)^2}(dX_1(t))^2 \\
&+ \frac{\partial f}{\partial X_2(t)}dX_2(t) + \frac{1}{2}\frac{\partial^2 f}{\partial X_2(t)^2}(dX_2(t))^2 + \frac{\partial f}{\partial X_1(t)\partial X_2(t)}dX_1(t)dX_2(t) \\
&= \left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial X_1(t)}a_1 + \frac{1}{2}\frac{\partial^2 f}{\partial X_1(t)^2}b_1^{\,2} + \frac{\partial f}{\partial X_2(t)}a_2 + \frac{1}{2}\frac{\partial^2 f}{\partial X_2(t)^2}b_2^{\,2} + \frac{\partial f}{\partial X_1(t)\partial X_2(t)}\rho b_1 b_2\right]dt \\
&+ \frac{\partial f}{\partial X_1(t)}b_1 dB_1(t) + \frac{\partial f}{\partial X_2(t)}b_2 dB_2(t)
\end{aligned}$$

多維隨機過程可表現如下:

$$\begin{aligned}
df(t, X_1(t), X_2(t), \ldots, X_n(t)) &= \frac{\partial f}{\partial t}dt + \sum_{i=1}^{n}\frac{\partial f}{\partial X_i(t)}dX_i(t) + \frac{1}{2}\sum_{i=1}^{n}\frac{\partial^2 f}{\partial X_i(t)^2}(dX_i(t))^2 \\
&+ \frac{1}{2}\sum_{i\neq j}\sum\frac{\partial f}{\partial X_i(t)\partial X_j(t)}dX_i(t)dX_j(t)
\end{aligned}$$

課堂練習:外國資產以本國貨幣計價

- 假定外國的資產爲X_1(以外國貨幣計價),匯率爲X_2(本國/外國)
  - 該資產以台幣計價的表示方式: $Y = X_1 X_2$
  - 假定 $dX_1 = a_1 X_1 dt + b_1 X_1 dB_1(t)$

    $dX_2 = a_2 X_2 dt + b_2 X_2 dB_2(t)$
  - 求算該資產的隨機過程表示式(dY)

數值模型

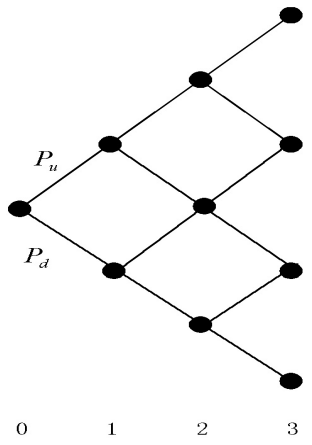
- 假定在市場上無套利機會,且market is complete
  - 存在唯一的風險中立機率 Q (見 第七和第八章)
- 在這種市場中,商品的評價可用取期望值的方式化簡
  - 假定有一個商品在到期日T時的報酬爲C(T),利率爲r
  - 在時間t時,該商品的價格爲: $e^{-r(T-t)}\mathrm{E}_{\mathrm{Q}}(\mathrm{C}(\mathrm{T}))$
  - 以一個履約價格爲X,標的物價格爲S(t)的賣權爲例
    - 賣權價格= $e^{-r(T-t)}\mathrm{E}_{\mathrm{Q}}((\mathrm{X}-\mathrm{S}(\mathrm{T}))^{+})$
- 建立一個數值模型,模擬在風險中立機率下,標的物價格的變化
  - 藉由該數值模型,可求出衍生物報酬的期望值→商品價格

數值模型的基本架構

- 假定標的物的隨機過程遵守對數常態分配
  - $dS(t) = \mu S(t)dt + \sigma S(t)dB(t)$
  - 經由<u>上述練習</u>,可知 $S(t) = S(0)e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma B(t)}$
- 在風險中立機率下,標的物的期望報酬率爲無風險利率r,所以上式中: $\mu = r$
- 討論兩種數值模型,模擬標的物價格變化
  - 二元樹模型
  - 蒙地卡羅法 (下一章介紹)

二元樹模型

- 回顧二元樹的架構:
  - 右圖爲三期的模型
  - 每一期價格只能上升或下降
  - 假定上升變 e^u 倍
    - 機率爲 P_u
  - 假定下降變 e^d 倍
    - 機率爲 P_d



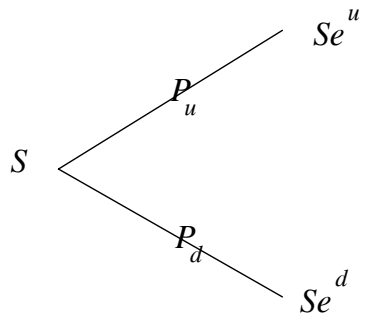
- 第n期的標的物價格可寫成 $S(0)e^{W(n)}$
  - $W(n) = I_1 + I_2 + ... + I_n$
  - $I_i = u$ or d

二元樹模型的建立

- 假定$I_1, I_2, ..., I_n$獨立,則根據中央極限定理,當 $n \to \infty$
  - W(n)會逼近常態分配 →二元樹逼近對數常態隨機過程
    →評價結果逼近理論價格
- 標的物價格過程:$S(T) = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma B(T)}$
  - $E(W_n) = \left(r - \frac{1}{2}\sigma^2\right)T$
  - $\mathrm{Var}(W_n) = \sigma^2 T$
- 所以 $I_1, I_2, ..., I_n$ 要滿足下列條件:
  - $\Delta t \equiv \frac{T}{n}$
  - $E(I_n) = \left(r - \frac{1}{2}\sigma^2\right)\Delta t$
  - $\mathrm{Var}(I_n) = \sigma^2 \Delta t$

二元樹模型的建立

- 考慮單期模型:
  - $P_u \times u + P_d \times d = \left(r - \frac{1}{2}\sigma^2\right)\Delta t \equiv m$
  - $P_u \times (u-m)^2 + P_d \times (d-m)^2 = \sigma^2 \Delta t$
  - $P_u + P_d = 1$

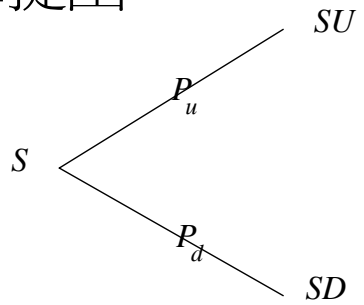


- 三個方程式,但是有四個未知數,可根據對評價的模型的要求加入不同等式
  - CRR model: $e^u \times e^d = 1$
    - Remark:在CRR模型設定中,爲了公式簡化,

$$P_u \times (u-m)^2 + P_d \times (d-m)^2 \to \sigma^2 \Delta t$$

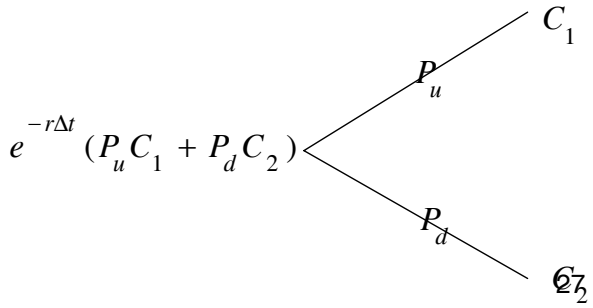
使用二元樹評價選擇權
CRR 二元樹模型

- 由 Cox, Ross, 和Rubinstein提出
- 模型設定如右
  - $U = e^{u} = e^{\sigma\sqrt{\Delta t}}$ $\quad D = e^{d} = e^{-\sigma\sqrt{\Delta t}}$



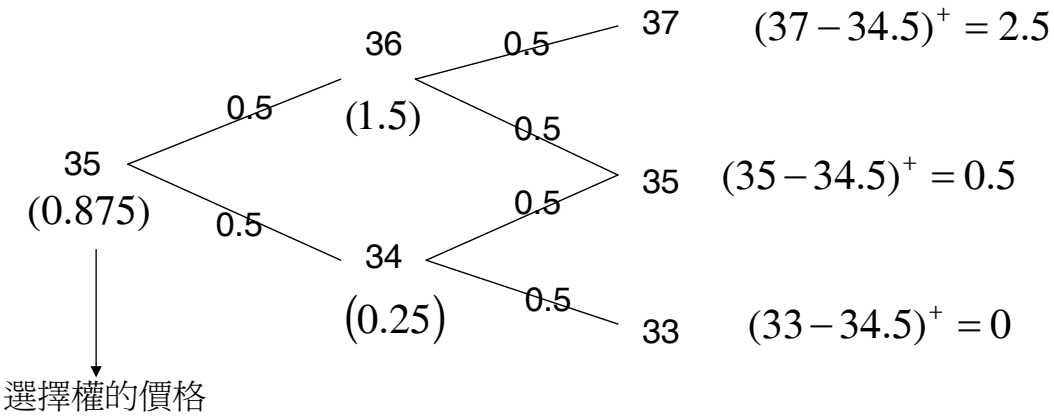
$$P_u = \frac{e^{r\Delta t} - D}{U - D} \qquad P_d = 1 - P_u$$

- 權利金的計算如右:



使用二元樹評價選擇權
Backward Induction

- 考慮兩期的模型 (令r=0)
  - 先計算最後一期的價格,然後再往前推



使用二元樹評價選擇權
Backward Induction

- 定義一個陣列,長度>(n+1)
- 計算最後一期的payoff,放入陣列中

2.5	0.5	0

- 利用第二期價格求算第一期價格

2.5	0.5	0

1.5	0.5	0

1.5	0.25	0

使用二元樹評價選擇權
Backward Induction

- 利用第一期的價格求算第零期的價格

1.5	0.25	0

0.875	0.25	0

- 選擇權的價格: 0.875

使用二元樹評價選擇權
程式流程設計

- 程式流程如下:
  - 1. 輸入資料: 標的物價格(S),標的物波動率(Sigma), 無風險利率(r),履約價格(X),時間長度(T),期數(n).
  - 2. 計算CRR二元樹的相關參數
  - 3. 求算選擇權在最後一期的payoff
  - 4. Backward Induction
  - 5. 輸出評價結果

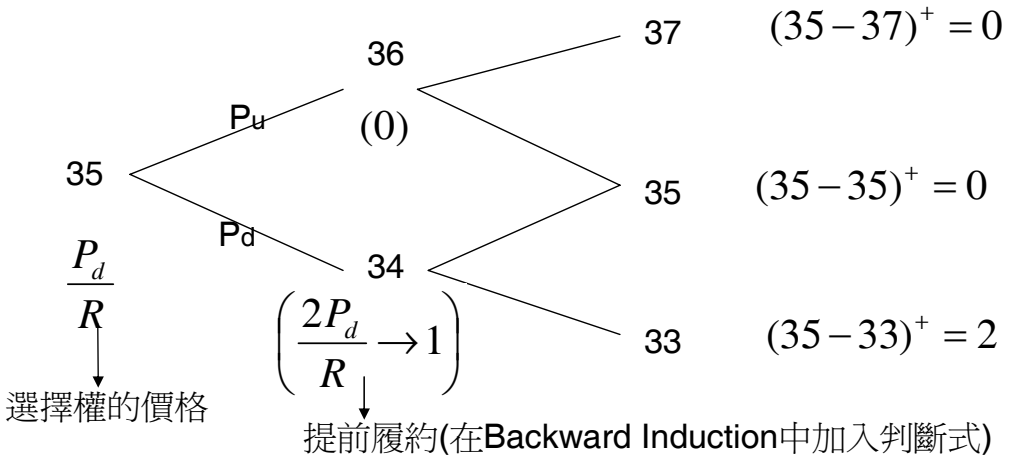
請參見C++財務程式設計 7.2

程式架構分析

```
void main()
{
 //輸入資料
 float S,T,X,r,Sigma;
 int n;  //期數
 printf("輸入標的物價格:");
 scanf("%f",&S);
 printf("輸入到期日:");
 scanf("%f",&T);
 printf("輸入履約價格:");
 scanf("%f",&X);
 printf("輸入無風險利率:");
 scanf("%f",&r);
 printf("輸入標的物價格波動率:");
 scanf("%f",&Sigma);
 printf("輸入期數:");
 scanf("%d",&n);
 //計算CRR二元樹的相關參數
 double U, D, Pu,Pd,DeltaT;
 DeltaT=T/n;
 U=exp(Sigma*sqrt(DeltaT));
 D=exp(-Sigma*sqrt(DeltaT));
 Pu=(exp(r*DeltaT)-D)/(U-D);
 Pd=1-Pu;
 //計算最後一期的payoff
 double Array[500];
 double CurrentS=S*pow(U,n);
 for(int i=0;i<=n;i=i+1)
 {
  Array[i]=Max(CurrentS-X,0);
  CurrentS=CurrentS*D*D;
 }
 //Backward Induction
 for(i=n-1;i>=0;i=i-1)
 {
  for(int j=0;j<=i;j++)
  {
   Array[j]=exp(-r*DeltaT)*(Array[j]*Pu+Array[j+1]*Pd);
  }
 }
 //輸出評價結果
 printf("選擇權價格=%f",Array[0]);
}
```

美式賣權評價

- 將上述程式改成評價賣權的程式
  - 賣權的payoff: $(X - S(T))^+$
- 考慮選擇權提前履約的問題(令X=35),一期利率爲R>1 (p>0.5)



Remark: 美式選擇權價格≥歐式選擇權

美式賣權評價
程式碼修改

```
// Backward Induction
for(i=n-1;i>=0;i=i-1)
   { CurrentS=S*pow(u,i);
    for(int j=0;j<=i;j++)
     {
      Array[j]=Max(exp(-r*DeltaT)*(Array[j]*p+Array[j+1]*(1-p)),X-CurrentS);
      CurrentS=CurrentS*d*d;
     }
    }
```

見 C++財務程式設計 7-2.6

Homework:

- C++財務程式設計 第七章 第六題和第八題

計算複雜度分析

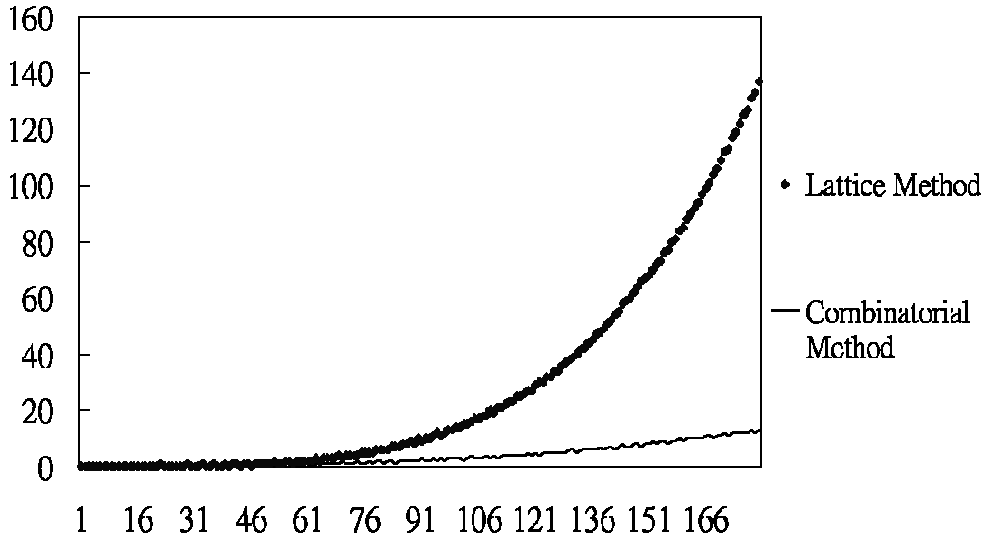
```
void main()
{
 //輸入資料
 float S,T,X,r,Sigma;
 int n;  //期數
 printf("輸入標的物價格:");
 scanf("%f",&S);
 printf("輸入到期日:");
 scanf("%f",&T);
 printf("輸入履約價格:");
 scanf("%f",&X);
 printf("輸入無風險利率:");
 scanf("%f",&r);
 printf("輸入標的物價格波動率:");
 scanf("%f",&Sigma);
 printf("輸入期數:");
 scanf("%d",&n);
 //計算CRR二元樹的相關參數
 double U, D, Pu,Pd,DeltaT;
 DeltaT=T/n;
 U=exp(Sigma*sqrt(DeltaT));
 D=exp(-Sigma*sqrt(DeltaT));
 Pu=(exp(r*DeltaT)-D)/(U-D);
 Pd=1-Pu;
 //計算最後一期的payoff
 double Array[500];
 double CurrentS=S*pow(U,n);
 for(int i=0;i<=n;i=i+1)
 {
  Array[i]=Max(CurrentS-X,0);
  CurrentS=CurrentS*D*D;
 }
 //Backward Induction
 for(i=n-1;i>=0;i=i-1)
 {
  for(int j=0;j<=i;j++)
  {
   Array[j]=exp(-r*DeltaT)*(Array[j]*Pu+Array[j+1]*Pd);
  }
 }
 //輸出評價結果
 printf("選擇權價格=%f",Array[0]);
}
```

淺藍色部分只需執行一次即可,
淺綠部分需執行n次
淺黃部分需執行n+(n-1)+(n-2)+…+1
共n(n+1)/2次

使用二元樹評價選擇權
計算複雜度分析

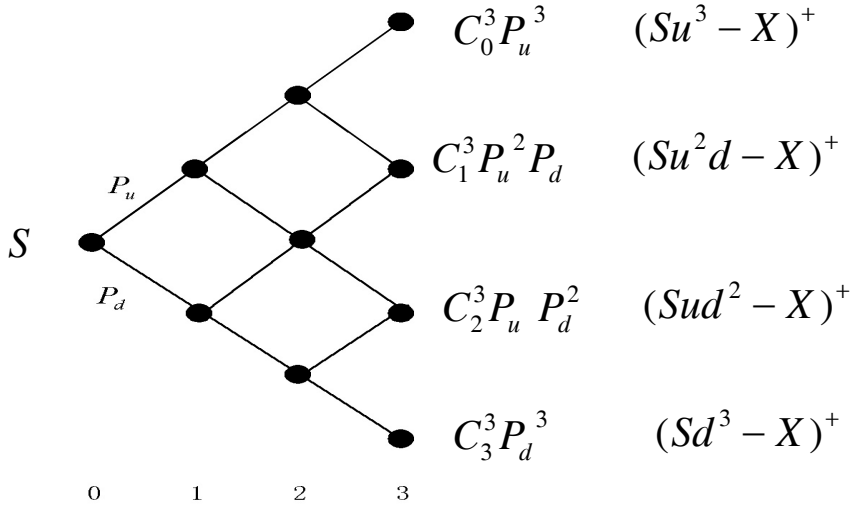
- $n \to \infty$,二元樹逼近對數常態隨機過程,評價結果會接近真實價值
- 分析n增大時,程式的複雜度變得十分重要
- 上述程式的計算時間在n變大時,主要受backward induction的影響
- 程式計算複雜度= $O(n^2)$
- 尋找一個能達到相同功能且計算複雜度較低的程式可提高計算效率

使用二元樹評價選擇權
計算時間的比較



使用組合數學
處理二元樹的評價問題

- 觀察一個三期的二元樹模型:



$$\text{選擇權價格} = e^{-rT} \sum_{i=0}^{3} C_i^3 P_u^{3-i} P_d^i (Su^{3-i} d^i - X)^+$$

使用組合數學
處理二元樹的評價問題

- 在n期的模型中,選擇權價格可表示如下:
  - $e^{-rT} \sum_{i=0}^{n} C_i^n P_u^{n-i} P_d^i (Su^{n-i} d^i - X)^+$
- 如果 $C_i^n P_u^{n-i} P_d^i (Su^{n-i} d^i - X)^+$ 可在固定時間內算出,則整個程式的計算複雜度可表現成O(n)
- 觀察第i個點和第i+1個點的機率和payoff,發現

$$C_i^n P_u^{n-i} P_d^i \times \left(\frac{n-i}{i+1} \times \frac{P_d}{P_u} \right) = C_{i+1}^n P_u^{n-i-1} P_d^{i+1}$$

$$Su^{n-i} d^i \times \frac{d}{u} = Su^{n-i-1} d^{i+1}$$

兩者皆可在固定時間內求出

使用組合數學
處理二元樹的評價問題

- 原來處理最後一期的報酬和backward induction的程式可縮減如下:

參照 8-2.2

```
double CurrentS=S*pow(U,n);
 double CurrentProb=pow(Pu,n);
 double OptionValue=0;
 for(int i=0;i<=n;i=i+1)
 {
  OptionValue=OptionValue+exp(-r*T)*CurrentProb*Max(CurrentS-X,0);
  CurrentS=CurrentS*D/U;
  CurrentProb=CurrentProb*(n-i)/(i+1)*Pd/Pu;
 }
```

- 淺藍區塊的程式只需執行n次,所以整個程式的計算複雜度降爲O(n)

課堂演練

- 討論是否可用組合數學的方式評價美式選擇權

- 考慮期數很大,程式是否能正常運作?
  - Overflow / Underflow problem
  - 使用log 運算

```
double a=pow(0.1,1000)*pow(0.1,-1000);.
printf("%lf",a);
```

換成

```
double a=exp(1000*log(0.1)-1000*log(0.1));
printf("%lf",a);
```

Homework

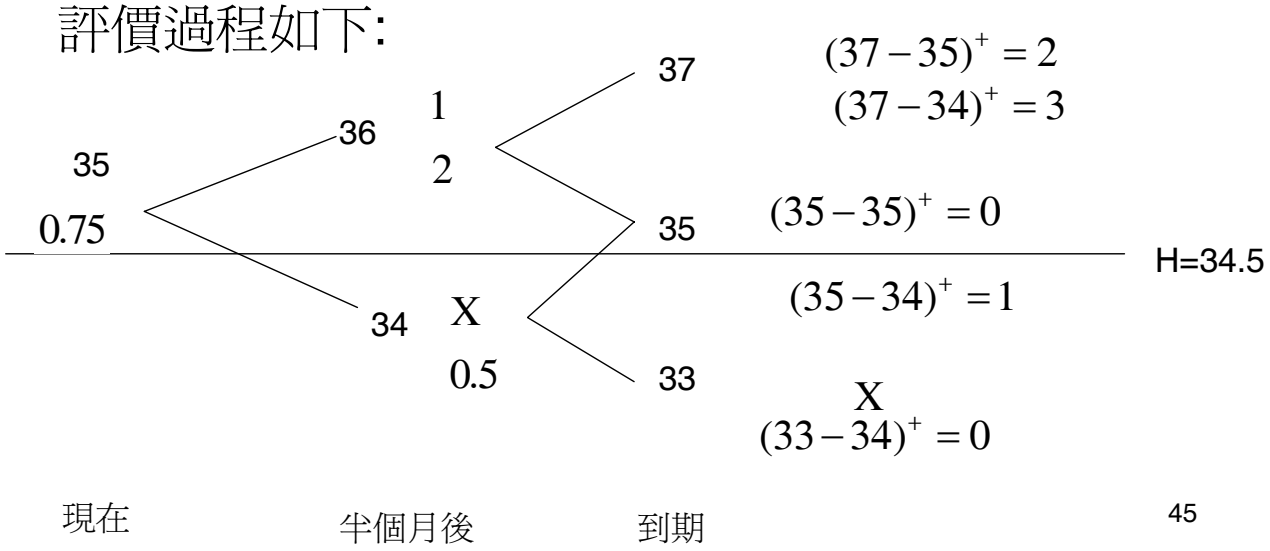
- Program Exercise：課本ch8習題3

在CRR樹上評價新奇選擇權

- 新奇選擇權的價格常會受標的物的價格路徑而影響
  - 可在每個節點上加上適當的狀態變數,來記憶不同狀況下的選擇權價格
  - 重設選擇權(Reset option):履約價格會在標的物價格碰到某一界限時重設
    - $$\text{payoff} = \begin{cases} (S(T)-X)^+ & \text{if } S(t)>H \quad \forall\, t\in(0,T) \\ (S(T)-B)^+ & \text{if } \exists\, t\in(0,T)\ S(t)\le H \end{cases}$$

重設選擇權評價

- 選擇權在時間t的價格,受到標的物價格在時間t之前是否碰觸預設價格H影響→每個節點最多需放入兩個狀態變數
- 以下圖爲例,假定H=34.5,重設的履約價格爲34,則評價過程如下:



評價程式撰寫

- 修改評價Vanilla option程式,用來評價重設選擇權
  - 增加輸入 B(重設後的履約價格), H(重設界限)
  - Array:改成二維陣列: Array[500][2];

    Array[*][0]→未重設的選擇權價格

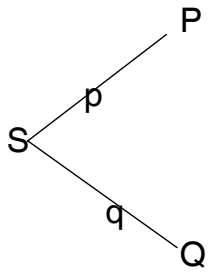
    Array[*][1]→重設後的選擇權價格
  - 最後一期的價格

    Array[i][0]=Max(CurrentS-X,0);

    Array[i][1]=Max(CurrentS-B,0);
  - 修改Backward induction
    - 分別考慮節點i的價格大於和小於H的處理 (See next slide)

評價程式撰寫

- Case 1: S<=H
  - Array[i][0] → Useless
  - Array[i][1]= $\frac{p\times V(P,1)+q\times V(Q,1)}{R}$
- Case 2: Q<=H<S
  - Array[i][0]= $\frac{p\times V(P,0)+q\times V(Q,1)}{R}$
  - Array[i][1]= $\frac{p\times V(P,1)+q\times V(Q,1)}{R}$
- Case 3:Otherwise
  - Array[i][0]= $\frac{p\times V(P,0)+q\times V(Q,0)}{R}$
  - Array[i][1]= $\frac{p\times V(P,1)+q\times V(Q,1)}{R}$



Code :最後一期報酬

```
double CurrentS=S*pow(u,n);
for(int i=0;i<=n;i=i+1)
{
 if(IsCall)
 {
     Array[i][0]=Max(CurrentS-X,0);
     Array[i][1]=Max(CurrentS-X1,0);
 }
 else
  {
     Array[i][0]=Max(X-CurrentS,0);
     Array[i][1]=Max(X1-CurrentS,0);
  }
  CurrentS=CurrentS*d*d;
}
```

Code: Backward Induction

```
for(i=n-1;i>=0;i=i-1)
{
 CurrentS=S*pow(u,i);
 for(int j=0;j<=i;j++)
 {
  double LowChildS=CurrentS*d;
    if(CurrentS<=L)
    {
                Array[j][1]=exp(-r*DeltaT)*(Array[j][1]*p+Array[j+1][1]*(1-p));
    }
    else if(LowChildS<=L)
    {
                Array[j][0]=exp(-r*DeltaT)*(Array[j][0]*p+Array[j+1][1]*(1-p));
                Array[j][1]=exp(-r*DeltaT)*(Array[j][1]*p+Array[j+1][1]*(1-p));
    }
    else
    {
       Array[j][0]=exp(-r*DeltaT)*(Array[j][0]*p+Array[j+1][0]*(1-p));
                Array[j][1]=exp(-r*DeltaT)*(Array[j][1]*p+Array[j+1][1]*(1-p));
    }
  CurrentS=CurrentS*d*d;
 }
}
```

障礙選擇權

- 選擇權是否失效(或生效)視標的物在到期日之前的價格是否曾經到達某障礙價格。
- 依標的物的價格碰到障礙價格而分
  - 出局選擇權(knock-out option)
  - 入局選擇權(knock-in option)
- 依標的物期初價格和障礙價格之關係而分
  - Up option
  - Down option
- 下出局選擇權(down-and-out option)

$$\text{payoff} = \begin{cases} (S(T)-X)^+ & \text{if } S(t)>L \quad \forall\, t\in(0,T) \\ 0 & \text{if } \exists\, t\in(0,T)\ \ S(t)\le L \end{cases}$$

Reduce Pricing of Barrier Options to the Pricing of Reset Options

- 下入局障礙買權(down-and-in barrier call option)：相當於 X設成∞, $X1$設成$X2$ 的重設選擇權。
- 下入局障礙賣權(down-and-in barrier put option)：相當於 X設成0, $X1$設成$X2$ 的重設選擇權。
- 下出局障礙買權(down-and-out barrier call option)：相當於X設成$X2$, $X1$設成∞的重設選擇權。
- 下出局障礙賣權(down-and-out barrier put option)：相當於 X設成$X2$, $X1$設成0 的重設選擇權。

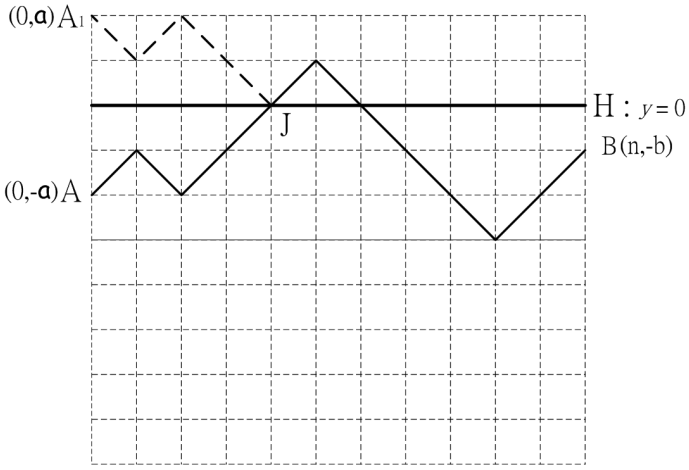
Homework

- 課本第七章習題7

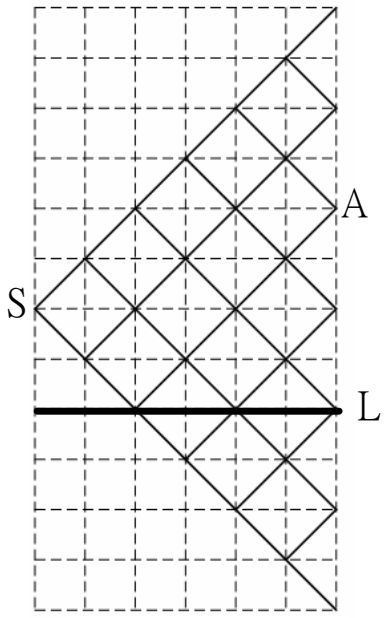
使用組合數學評價障礙選擇權

- 使用格子樹模型評價: $O(n^2)$
- 考慮如何使用組合數學加速評價 O(n)
- 反射原理:

$$\begin{cases} \alpha + \beta = n \\ \beta - \alpha = a + b \end{cases}$$



利用反射原理評價 Down-and-in Single Barrier Option

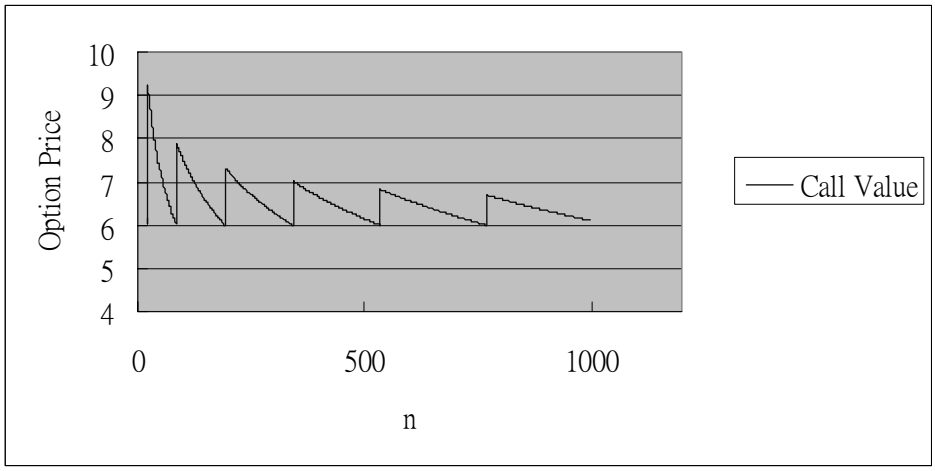


Homework:

- 修改8-2.5 的程式來評價 Down-and out single barrier option.

鋸齒狀的收斂行爲

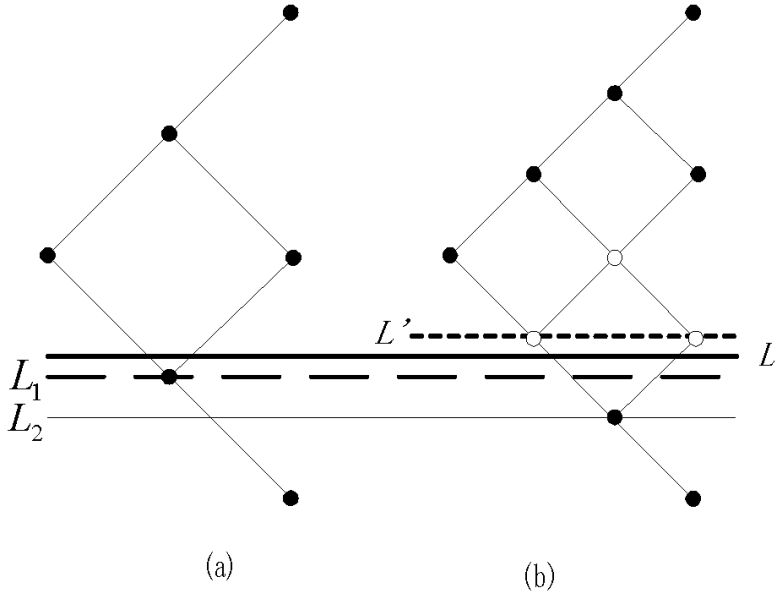
- 以CRR來評價障礙選擇權 (S=95,X=100,T=1,volitility=25%,r=10%,H=90)



- 如何改善？
  - 誤差來自何處？

使用二元樹模型評價造成誤差原因

- 有效障礙隨n而改變



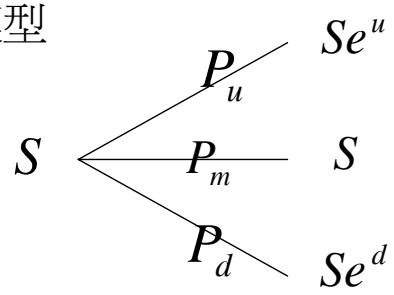
如何建構一個樹狀結構,使得有效障礙不隨n而改變?

使用Kamrad & Ritchken 三元樹模型評價障礙選擇權

- 由Kamrad and Ritchken [1991] 提出
- 是一個擁有延伸參數 $\lambda(\geq 1)$ 的三元樹模型
- 模型的設定：
  - $e^{u} = e^{\lambda\sigma\sqrt{\Delta t}}$

    $e^{m} = e^{0} = 1$

    $e^{d} = e^{-\lambda\sigma\sqrt{\Delta t}}$



- 標的物價格過程: $S(T) = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma B(T)}$
  - 由三元樹來模擬標的物的價格→三元樹的期望值和變異數要符合標的物價格過程的期望值和變異數
  - $p_u(\lambda\sigma\sqrt{\Delta t}) + p_d(-\lambda\sigma\sqrt{\Delta t}) = (r - 1/2\sigma^2)\Delta t$

    $p_u(\lambda\sigma\sqrt{\Delta t})^2 + p_d(-\lambda\sigma\sqrt{\Delta t})^2 = \sigma^2\Delta t$

    兩個變數 p_u, p_d ,兩個方程式

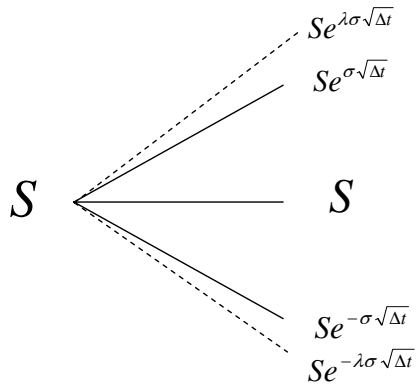
使用Kamrad & Ritchken 三元樹模型評價障礙選擇權

- 再加上 $p_u + p_m + p_d = 1$ 條件
  - 可得到 $(\alpha \equiv r - \frac{1}{2}\sigma^2)$

$$P_u = \frac{1}{2\lambda^2} + \frac{\alpha\sqrt{\Delta t}}{2\lambda\sigma}$$

$$P_m = 1 - \frac{1}{\lambda^2}$$

$$P_d = \frac{1}{2\lambda^2} - \frac{\alpha\sqrt{\Delta t}}{2\lambda\sigma}$$



- Ritchken[1995] 透過Kamrad & Ritchken三元樹模型中參數 λ 的適當選取，使三元樹模擬的標的物價格S(t)能恰恰好落在障礙價格H上

使用Rtichken三元樹評價下出局障礙買權程式流程設計

- 程式流程如下:
  - 1. 輸入資料: 標的物價格(S),標的物波動率(Sigma), 無風險利率(r),履約價格(X),時間長度(T),期數(n).
  - 2. 計算Ritchken三元樹的相關參數
    - Lambda：延伸參數
    - Lindex：從標的物初使價格往下走幾步到障礙
  - 3. 求算選擇權在最後一期的payoff
    - 計算至落在障礙的節點(第n+Lindex個)即可停止
  - 4. Backward Induction
    - 每次均計算至落在障礙的節點
  - 5. 輸出評價結果

請參見7-4.4

Ritchken project

下出局障礙買權

```
void main()
{
  //輸入資料
  float S,T,X,r,Sigma;
  int n;  //期數
  printf("輸入標的物價格:");
  scanf("%f",&S);
  printf("輸入到期日:");
  scanf("%f",&T);
  printf("輸入履約價格:");
  scanf("%f",&X);
  printf("輸入無風險利率:");
  scanf("%f",&r);
  printf("輸入標的物價格波動率:");
  scanf("%f",&Sigma);
  printf("輸入期數:");
  scanf("%d",&n);

//計算相關參數
  double dt=T/n;
  double tmp = log(S/L)/(Sigma*sqrt(dt));
  int Lindex = (int)floor(tmp);
  double lambda=tmp/Lindex;

  double R=exp(r*dt);
  double U=exp(lambda*Sigma*sqrt(dt));
  double D=1/U;
  double pu=1/(2*lambda*lambda)+
(r-Sigma*Sigma/2.0)*sqrt(dt)/(2*lambda*Sigma);
  double pd=1/(2*lambda*lambda)-
(r-Sigma*Sigma/2.0)*sqrt(dt)/(2*lambda*Sigma);
  double pm=1.0-pu-pd;

//計算最後一期的payoff
double Array[1000];
double CurrentS=S*pow(U,n);
for(int i=0;i<n+Lindex;i++)
{
   Array[i]=Max(CurrentS-X,0);
   CurrentS=CurrentS*D;
}
Array[n+Lindex]=0;

  //Backward Induction
for(int j=n-1;j>=0;j--)
{   int i=0;
    for(i=0;i<Min(j+Lindex,2*j+1);i++)
    {
     Array[i]=(pu*Array[i]+pm*Array[i+1]+pd*Array[i+2])/R
    }
    if(i==j+Lindex)      Array[i] = 0;
 }
  //輸出評價結果
  printf("障礙選擇權價格=%f",Array[0]);
}
```


Homework

- 輸入不同的n,判斷本程式的收斂
- 修改本程式來評價一上出局障礙買權