長天期利率衍生商品之評價(三)

利率期限結構模型的傳統方法

晌騰金融科技

技術長

董夢雲 博士

dongmy@ms5.hinet.net

目　錄

Part V 鞅性訂價與傳統單因子利率模型

Part V 鞅性訂價與傳統單因子

利率模型

一、單因子利率模型

(一)基本數學關係

◆ 即期利率與債券價格

- 令 P(t, s)為 s 時點到期支付一元之債券(pure discount bond)，在 t 時點的市場價格。
- R(t, s)為 t 時點上，s 時點到期的連續複利的即期利率(continuously compound spot rate)。

$$P(t,s) = Exp\left[-R(t,s)\times(s-t)\right]$$

$$R(t,s) = -\frac{1}{s-t}\ln\left[P(t,s)\right]$$

◆ 遠期利率與債券價格

- 令 P(t, s)為 s 時點到期支付一元之債券，在 t 時點的市場價格。
- f(t, s)為 t 時點上，s 時點到期的瞬間遠期利率(instantaneous forward rate)。

$$f(t,s) = -\frac{\partial}{\partial s}\ln\left[P(t,s)\right]$$

$$P(t,s) = Exp\left[-\int_t^s f(t,\tau)\cdot d\tau\right]$$

◆ 遠期利率與即期利率

- 即期利率為遠期利率的連續平均，

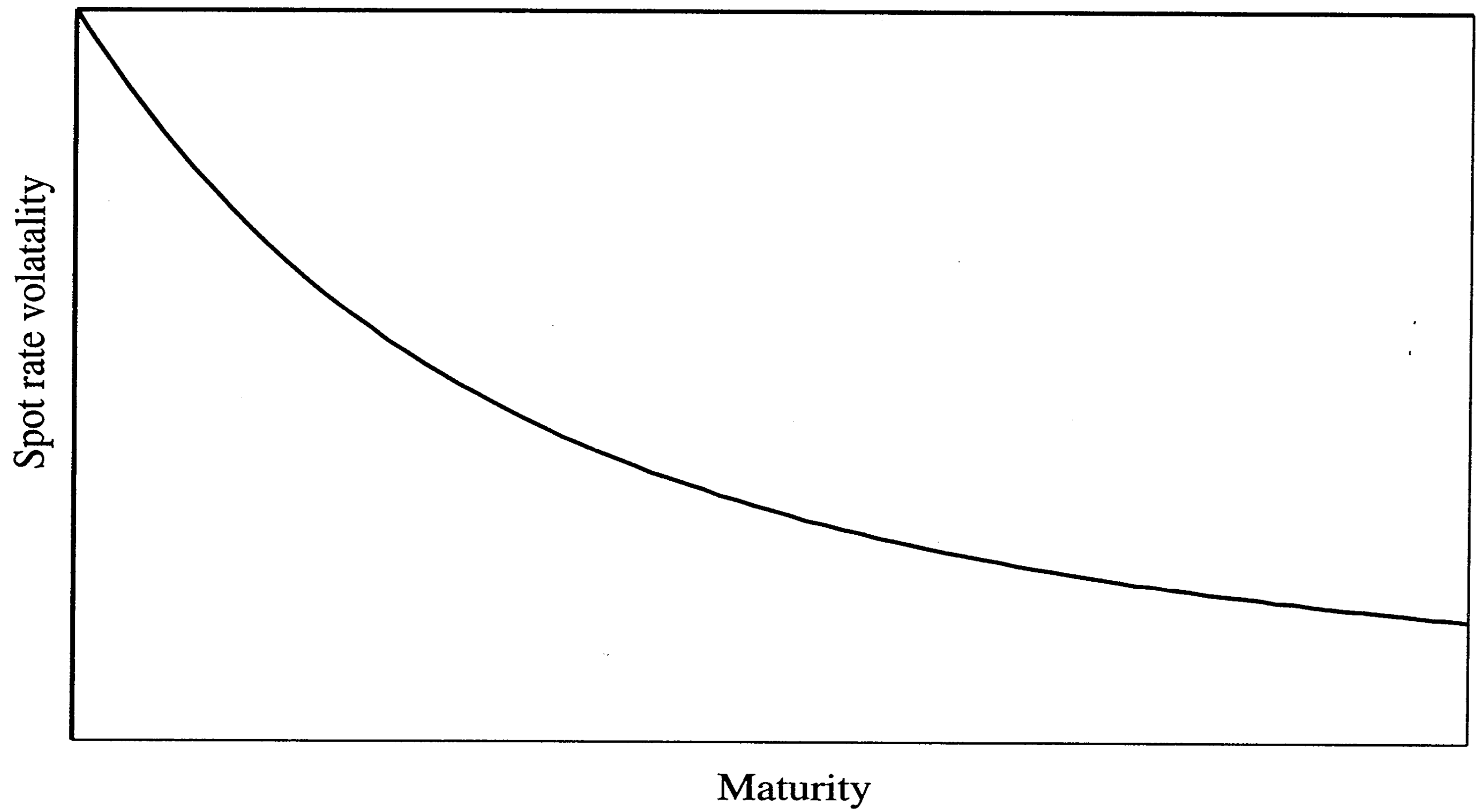
$$R(t,s)=\frac{1}{s-t}\left[\int_t^s f(t,\tau)\cdot d\tau\right]$$

- 令 r(t)表 t 時點上的短期利率(short-term interest rate)，r(t) = R(t, t)。
- $\sigma_R(t,s)$ 為即期利率 R(t, s)的波動性，$\sigma(r)$ 為短期利率 r(t)的波動性，

$$\sigma(r)=\sigma_R(t,t)$$

◆ Spot Rate Volatility Structure

FIGURE 6.1 Spot Rate Volatility Structure



◆ Notation Summary：

$P(t, s)$	$=$ price at time t of a pure discount bond that matures at time s
$R(t, s)$	$=$ yield at time t on the s-maturity pure discount bond (spot rate)
r	$=$ short-term interest rate
$f(t, s)$	$=$ instantaneous forward rate at time t for time s
K	$=$ strike price of the option
$c(t, T, s)$	$=$ price at time t of a European call option with exercise date T on an s-maturity pure discount bond ($t \le T \le s$) and with strike price K
$p(t, T, s)$	$=$ price at time t of a European put option with exercise date T on an s-maturity pure discount bond ($t \le T \le s$) and with strike price K
$\sigma(r)$	$=$ volatility of the short rate
$\sigma_R(t, s)$	$=$ volatility of yield $R(t, s)$

(二) Vasicek模型

- Vasicek(1977)與 CIR(1985)以瞬間短期利率為唯一的不確定性來源，Vasicek 以下式來描述短期利率的變動，

$$dr_t = \alpha(\bar{r} - r_t) \cdot dt + \sigma \cdot dZ_t \quad \text{(1.1)}$$

  - 其中 α、σ 均為非負的常數，
    - ✓ 為一均數復歸的程序，
    - ✓ 稱之為 Ornstein-Uhlenbeck 程序。
  - 此一程序為真實世界的機率測度，可將之轉換到風險中立的機率測度。

◆ 可以證明得知，在 Vasicek 模型中，真實世界中未來的短期利率為常態分配的，其平均數與變異數分別為

$$E_t[r_T]=\bar{r}+(r_t-\bar{r})e^{-\alpha(T-t)} \quad \text{.......(1.2)}$$

$$Var_t[r_T]=\frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha(T-t)}\right) \quad \text{.......(1.3)}$$

- 當 T→∞時，平均數趨近$\bar{r}$，與變異數趨近$\sigma^2/(2\alpha)$。
- 當 $\alpha\rightarrow\infty$時，平均數趨近$\bar{r}$，與變異數趨近 0。
- 當 $\alpha\rightarrow 0$ 時，平均數趨近r_t，與變異數趨近$\sigma^2(T-t)$。
- 目前短期利率與長期利率的差距，在 ln(2)/α 後，減為一半。

(三) CIR模型

- Cox, Ingersoll, & Ross(1985)以下式來描述短期利率的變動，

$$dr_t = \alpha(\bar{r} - r_t)\cdot dt + \sigma\sqrt{r_t}\,dZ_t \quad \text{..........(1.4)}$$

- 其中 α、σ 均為非負的常數，
  - ✓ 為一均數復歸的程序，
  - ✓ 利率的波動性為利率的大小增函數，
  - ✓ 利率不會產生負值。
- 此一程序為真實世界的機率測度，可將之轉換到風險中立的機率測度。

◆ 可以證明得知，在 CIR 模型中，真實世界中未來的短期利率為 Non-central χ^2 分配，其平均數與變異數分別為

$$E_t[r_T] = \bar{r} + (r_t - \bar{r})e^{-\alpha(T-t)} \quad \text{.......(1.5)}$$

$$Var_t[r_T] = \frac{\sigma^2 r_t}{\alpha}\left(e^{-\alpha(T-t)} - e^{-2\alpha(T-t)}\right) + \frac{\sigma^2 \bar{r}}{2\alpha}\left(1 - e^{-\alpha(T-t)}\right)^2 \quad \text{.......(1.6)}$$

- 當 T→∞時，平均數趨近 $\bar{r}$ ，與變異數趨近 $\sigma^2 \bar{r}/(2\alpha)$ 。
- 當 α→∞時，平均數趨近 $\bar{r}$ ，與變異數趨近 0。
- 當 α→0 時，平均數趨近 r_t，與變異數趨近 $\sigma^2 r_t (T-t)$ 。

二、鞅性定價理論

(一)風險中立鞅性測度

- 根據風險中立定價理論，一資產的價格程序 P_t ，有下面的關係

$$P_t = E_t^Q\left[e^{-\int_t^s r_u du} P_s \right]，t < s \le T$$

  - 其中 Q 表一風險中立的機率測度，代表價格的折現程序為 Q-martingale。
  - 針對特定資產的價格，則由其風險中立之測度與償付來決定。

- 針對 T 時點到期的零息債券，在 t 時點的價格可寫為

$$B_t^T = E_t^Q\left[e^{-\int_t^T r_u du} \right]，t < T$$

  - 因為 $B_T^T = 1$，正確的解需由風險中立下的利率程序去計算。

◆ 針對 T 時點有單一償付 H_T 的證券，在 t 時點的價格可寫為

$$P_t = E_t^Q\left[e^{-\int_t^T r_u du} H_T\right]，t < T$$

- 假設利率程序與償付數量皆為擴散程序 x_t 與時間的函數。

$$r_t = r(x_t, t)\ ，\ H_T = H(x_T, T)$$

- 擴散程序 x_t 的動態由下式表示之。

$$dx_t = \alpha(x_t, t)dt + \beta(x_t, t)dZ_t$$

- 令風險性資產之 P_t 的動態可以下式表示，

$$dP_t = P_t[\mu_t dt + \sigma_t dZ_t]$$

  - 其折現價格可表示為，

$$\overline{P}_t = P_t e^{-\int_0^t r_u du}$$

- 根據 Ito's Lemma，折現價格的動態可以下式表示

$$d\overline{P}_t = \overline{P}_t\left[(\mu_t - r_t)dt + \sigma_t dZ_t\right]$$

  - 令風險中立 Q 測度下的 Wiener 程序為

$$dZ_t^Q = dZ_t + \lambda_t dt \quad \text{..........(2.1)}$$

  - 則 Q 測度下的折現價格動態為

$$d\overline{P}_t = \overline{P}_t\left[(\mu_t - r_t - \sigma_t\lambda_t)dt + \sigma_t dZ_t^Q\right]$$

- 如果折現價格在 Q 測度下為鞅性的，則其漂移項必須為零，

$$\mu_t - r_t - \sigma_t \lambda_t = 0$$

  - λ_t 稱之為市場風險的價格

$$\lambda_t = \frac{\mu_t - r_t}{\sigma_t}$$

  - Q 測度與現有 P 測度間的關連，可由 Radon-Nikodym Derivative 表示

$$\frac{dQ}{dP} = \xi_t = \exp\left\{ -\int_0^t \lambda_u dZ_u - \frac{1}{2}\int_0^t \lambda_u^2 du \right\}$$

- 對任一隨機程序 $x = (x_t)_{t \in [0,T]}$ ，

$$E_t^Q[x_s] = E_t^P\left[\frac{\xi_s}{\xi_t} x_s\right] = E_t^P\left[x_s \exp\left\{ -\int_t^s \lambda_u dZ_u - \frac{1}{2}\int_t^s \lambda_u^2 du \right\}\right] \text{，} 0 \le t < s \le T$$

(二)一般化鞅性測度

◆ 事實上，風險中立的 Q 測度，是以 Bank Account 為計價單位的鞅性測度，

$$\overline{P}_t = P_t e^{-\int_0^t r_u du} = \frac{P_t}{e^{\int_0^t r_u du}} = \frac{P_t}{A_t} \text{ , } P_0 = \frac{P_0}{A_0} = E_0^Q\left[\frac{P_t}{A_t}\right] = E_0^Q\left[\frac{P_t}{e^{\int_0^t r_u du}}\right]$$

➢ 我們可依據需要，選用適當的計價單位與鞅性測度。

◆ 例如，我們可以選擇配合以資產價格 S_t 為計價單位的鞅性測度 Q^S ，使得要計算的資產價格 P_t ，滿足下面的關係，

$$\frac{P_t}{S_t} = E_t^{Q^S}\left[\frac{P_T}{S_T}\right]$$

➢ 通常，這是由於 P_T / S_T 的分配，在 Q^S 下相對的簡單，使得以 S_t 為計價單位的 P_t 求算，呈現出計算上的便利性。

◆ 令資產價格 S_t 與 P_t 分別有下面的動態

$$dS_t = S_t[\mu_{S_t} dt + \boldsymbol{\sigma}_{S_t}^{\mathrm{T}} d\mathbf{Z}_t]$$

$$dP_t = P_t[\mu_{P_t} dt + \boldsymbol{\sigma}_{P_t}^{\mathrm{T}} d\mathbf{Z}_t]$$

➢ 其中，$d\mathbf{Z}_t$ 為一行向量，表隨機性的干擾源。

◆ 根據 Ito's Lemma，可得

$$d\left(\frac{P_t}{S_t}\right) = \frac{P}{S}\left[(\mu_{P_t} - \mu_{S_t} + \left\|\boldsymbol{\sigma}_{S_t}\right\|^2 - \boldsymbol{\sigma}_{S_t}^{\mathrm{T}}\boldsymbol{\sigma}_{P_t})dt + (\boldsymbol{\sigma}_{P_t} - \boldsymbol{\sigma}_{S_t})^{\mathrm{T}} d\mathbf{Z}_t\right]$$

➢ Q^S 測度下的標準布朗運動定義如下

$$d\mathbf{Z}_t^S = d\mathbf{Z}_t + \lambda_t^S dt \quad \text{..........(2.2)}$$

➢ 如下選擇 λ_t^S ，

$$(\boldsymbol{\sigma}_{P_t} - \boldsymbol{\sigma}_{S_t})^{\mathrm{T}} \lambda_t^S = (\mu_{P_t} - \mu_{S_t} + \left\|\boldsymbol{\sigma}_{S_t}\right\|^2 - \boldsymbol{\sigma}_{S_t}^{\mathrm{T}}\boldsymbol{\sigma}_{P_t})$$

◆ 可以得到Q^S下的鞅性測度

$$d\left(\frac{P_t}{S_t}\right)=\frac{P}{S}\left[(\boldsymbol{\sigma}_{P_t}-\boldsymbol{\sigma}_{S_t})^{\mathrm{T}}d\mathbf{Z}_t^S\right]$$

➢ λ_t^S 與 λ_t 有下面的關係

$$\boldsymbol{\lambda}_t^S=\boldsymbol{\lambda}_t-\boldsymbol{\sigma}_{S_t} \quad \text{......(2.3)}$$

$$(\boldsymbol{\sigma}_{P_t}-\boldsymbol{\sigma}_{S_t})^{\mathrm{T}}\boldsymbol{\lambda}_t^S=(\mu_{P_t}-\mu_{S_t}+\left\|\boldsymbol{\sigma}_{S_t}\right\|^2-\boldsymbol{\sigma}_{S_t}^{\mathrm{T}}\boldsymbol{\sigma}_{P_t})$$

$$\mu_{P_t}=r_t+\boldsymbol{\sigma}_{P_t}^{\mathrm{T}}\boldsymbol{\lambda}_t$$

$$\mu_{S_t}=r_t+\boldsymbol{\sigma}_{S_t}^{\mathrm{T}}\boldsymbol{\lambda}_t$$

$$\left\|\boldsymbol{\sigma}_{S_t}\right\|^2=\boldsymbol{\sigma}_{S_t}^{\mathrm{T}}\boldsymbol{\sigma}_{S_t}$$

$$(\boldsymbol{\sigma}_{P_t}-\boldsymbol{\sigma}_{S_t})^{\mathrm{T}}\boldsymbol{\lambda}_t^S=(\boldsymbol{\sigma}_{P_t}-\boldsymbol{\sigma}_{S_t})^{\mathrm{T}}(\boldsymbol{\lambda}_t-\boldsymbol{\sigma}_{S_t})$$

(三)遠期鞅性測度

◆ 針對只在單一時點 T 有償付的衍生商品，通常會選用以 T 時點到期的零息債券為其計價單位。

➢ 令 T 時點到期的零息債券，在 t 時點的價格為 B_t^T，其動態為

$$dB_t^T = B_t^T[(r_t + (\boldsymbol{\sigma}_t^T)^{\mathrm{T}}\boldsymbol{\lambda}_t)dt + (\boldsymbol{\sigma}_t^T)^{\mathrm{T}} d\mathbf{Z}_t]$$

➢ 則在 T 時點支付 P_T 的資產，在 t 時點的價格可如下求得

$$\frac{P_t}{B_t^T} = E_t^{Q^T}\left[\frac{P_T}{B_T^T}\right] = E_t^{Q^T}[P_T]$$

➢ 由前一節一般化鞅性測度可知，機率測度 Q^T 下的布朗運動為

$dZ_t^Q = dZ_t + \lambda_t dt$，由(2.1)，

$dZ_t^S = dZ_t + \lambda_t^S dt$，由(2.2)，$\lambda_t^T = \lambda_t - \sigma_t^T$，由(2.3)

$dZ_t^T = dZ_t + \lambda_t^T dt = dZ_t + (\lambda_t - \sigma_t^T)dt$，

$dZ_t^T = dZ_t^Q - \sigma_t^T dt$

三、利率程序的模擬

(一)亂數的產生

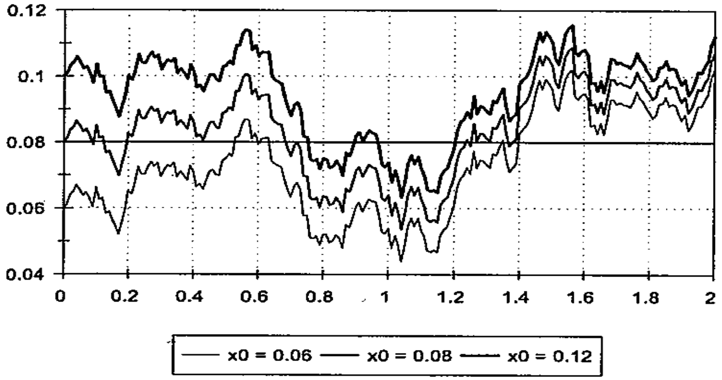
- 利用系統所附的 U[0,1]亂數，配合常態分配累積機率密度函數的反函數，產生常態分配的亂數。
  - U = Rnd()：產生 Uniform[0,1]的亂數
  - N = NormSDist()：常態分配累積機率密度函數
  - N^{-1} = NormSInv()：常態分配累積機率密度函數的反函數

(二)Vasicek模型的模擬

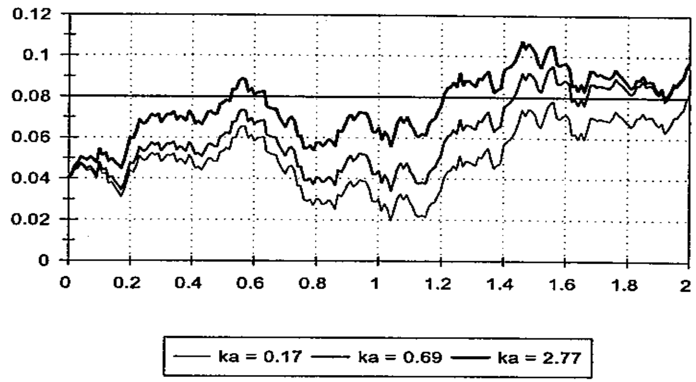
- 由(1.1)式可得下面的模擬方程式，

$$r_t = r_{t-1} + \alpha(\bar{r} - r_{t-1})(\Delta t) + \sigma\varepsilon_t\sqrt{\Delta t} \quad \text{...}(3.1)$$

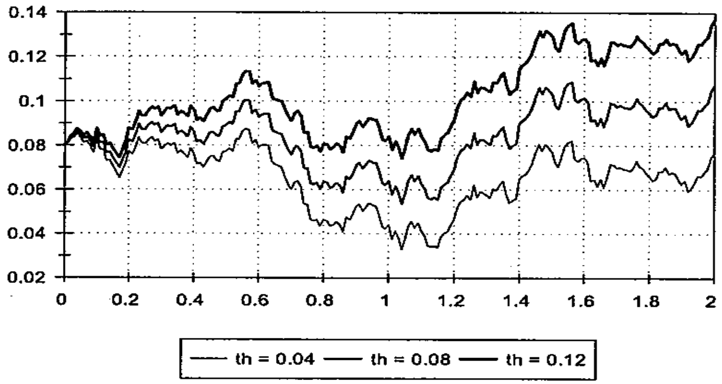
  - 此模擬式之產出變數為常態分配之隨機變數。
  - 近似的模擬方法。可以直接用(1.2)與(1.3)的結果，產生正確的模擬。



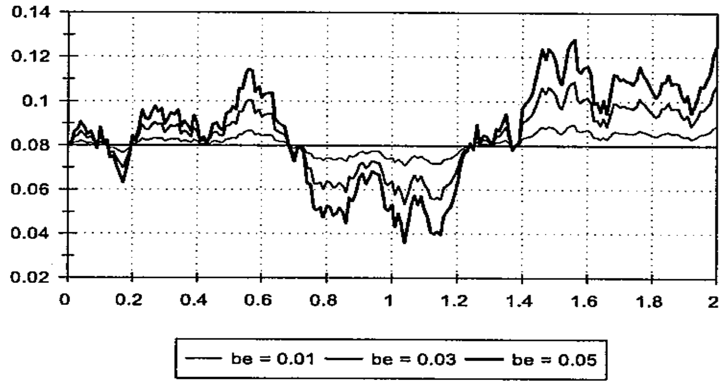
(a) Different initial values x_0



(b) Different κ-values; $x_0 = 0.04$

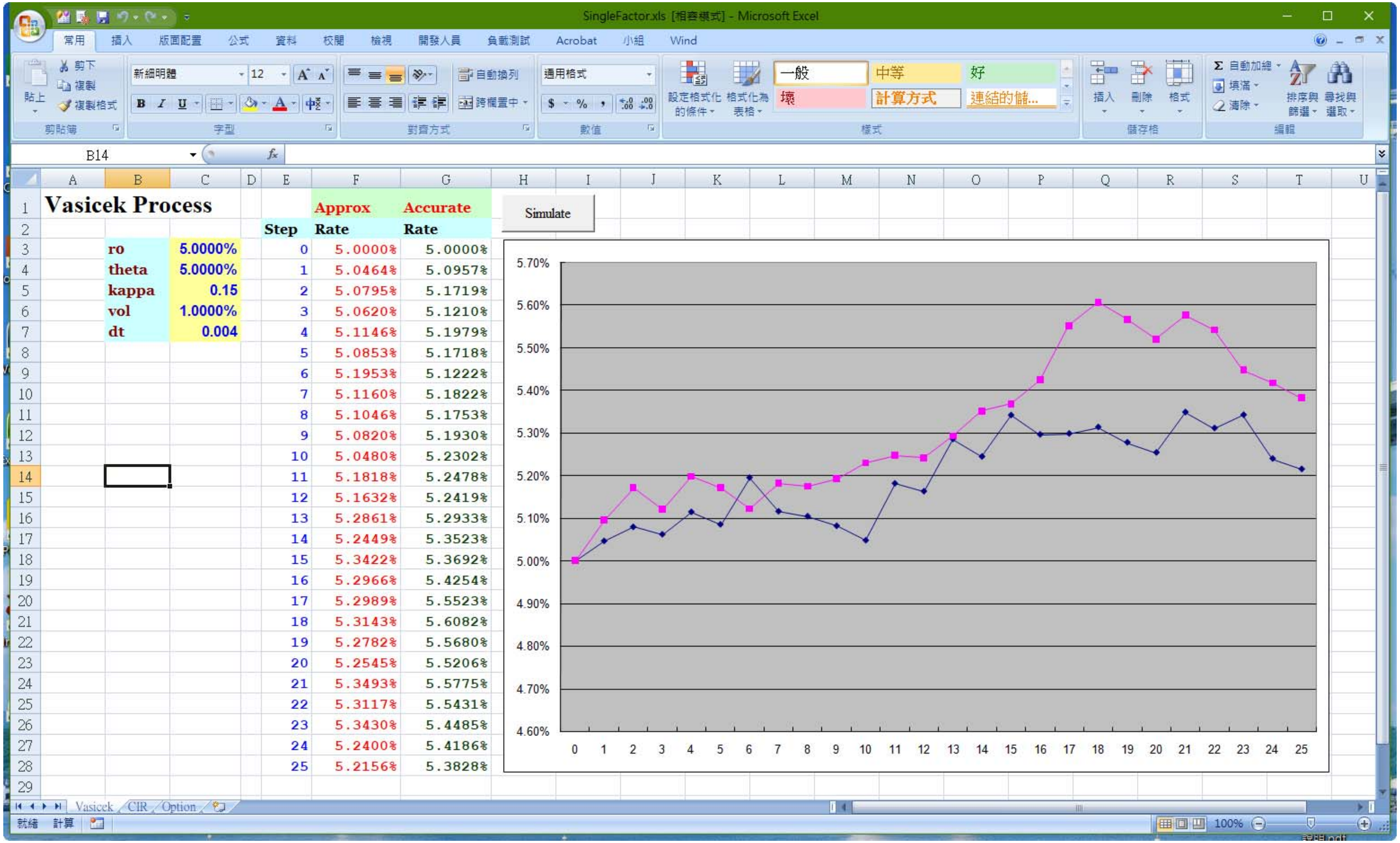


(c) Different θ-values



(d) Different β-values

Figure 3.5: Simulated paths for an Ornstein-Uhlenbeck process. The basic parameter values are $x_0 = \theta = 0.08$, $\kappa = \ln 2 \approx 0.69$, and $\beta = 0.03$.



Vasicek Process

ro	5.0000%
theta	5.0000%
kappa	0.15
vol	1.0000%
dt	0.004

Step	Approx Rate	Accurate Rate
0	5.0000%	5.0000%
1	5.0464%	5.0957%
2	5.0795%	5.1719%
3	5.0620%	5.1210%
4	5.1146%	5.1979%
5	5.0853%	5.1718%
6	5.1953%	5.1222%
7	5.1160%	5.1822%
8	5.1046%	5.1753%
9	5.0820%	5.1930%
10	5.0480%	5.2302%
11	5.1818%	5.2478%
12	5.1632%	5.2419%
13	5.2861%	5.2933%
14	5.2449%	5.3523%
15	5.3422%	5.3692%
16	5.2966%	5.4254%
17	5.2989%	5.5523%
18	5.3143%	5.6082%
19	5.2782%	5.5680%
20	5.2545%	5.5206%
21	5.3493%	5.5775%
22	5.3117%	5.5431%
23	5.3430%	5.4485%
24	5.2400%	5.4186%
25	5.2156%	5.3828%

```
Public Sub Vasicek_Test()
    Dim I As Integer
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim r1 As Double
    Dim SporRate As Double

    r0 = Worksheets("Vasicek").Range("C3").Value
    rbar = Worksheets("Vasicek").Range("C4").Value
    alpha = Worksheets("Vasicek").Range("C5").Value
    sig = Worksheets("Vasicek").Range("C6").Value
    DT = Worksheets("Vasicek").Range("C7").Value

    Call Vasicek_InitObj(r0, rbar, alpha, sig, DT)

    Call Vasicek_GetBondParameters(0, 1)
    'MsgBox VS.ZeroBondPrice

    'Rnd (-4)
    For I = 1 To 25
        r1 = Vasicek_GetNextRateByApprox
        Worksheets("Vasicek").Cells(I + 3, 6).Value = r1
        VS.InitRate = r1
    Next I

    Call Vasicek_InitObj(r0, rbar, alpha, sig, DT)
    'Rnd (-4)
    For I = 1 To 25
```

```
        r1 = Vasicek_GetNextRate
        Worksheets("Vasicek").Cells(I + 3, 7).Value = r1
        VS.InitRate = r1
    Next I

End Sub
```

```
Public Function Vasicek_GetNextRateByApprox() As Double
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim dZ As Double
    Dim r1 As Double

    r0 = VS.InitRate
    rbar = VS.LongRate
    alpha = VS.AdjSpeed
    sig = VS.Volatility
    DT = VS.DeltaTime
    dZ = Sqr(DT) * Application.NormSInv(Rnd)

    r1 = r0 + alpha * (rbar - r0) * DT + sig * dZ
    Vasicek_GetNextRateByApprox = r1
End Function
```

```
Public Function Vasicek_GetNextRate() As Double
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim dZ As Double
    Dim mean As Double
    Dim std As Double
    Dim r1 As Double

    r0 = VS.InitRate
    rbar = VS.LongRate
    alpha = VS.AdjSpeed
    sig = VS.Volatility
    DT = VS.DeltaTime
    dZ = Application.NormSInv(Rnd)
    mean = rbar + (r0 - rbar) * Exp(-alpha * DT)
    std = Sqr(((sig * sig) / (2 * alpha)) * (1 - Exp(-2 * alpha * DT)))

    r1 = mean + std * dZ
    Vasicek_GetNextRate = r1
End Function
```

```vba
'******************* Object Module ******************************
Type VasicekClass
    InitRate As Double
    LongRate As Double
    AdjSpeed As Double
    Volatility As Double
    DeltaTime As Double
    SpotRate As Double
    SpotVol As Double
    ZeroBondPrice As Double
End Type

Public VS As VasicekClass


'******************* Vasicek Module ******************************
Public Sub Vasicek_InitObj(r0, rbar, alpha, sig, DT)
    VS.InitRate = r0
    VS.LongRate = rbar
    VS.AdjSpeed = alpha
    VS.Volatility = sig
    VS.DeltaTime = DT
    Randomize
End Sub
```

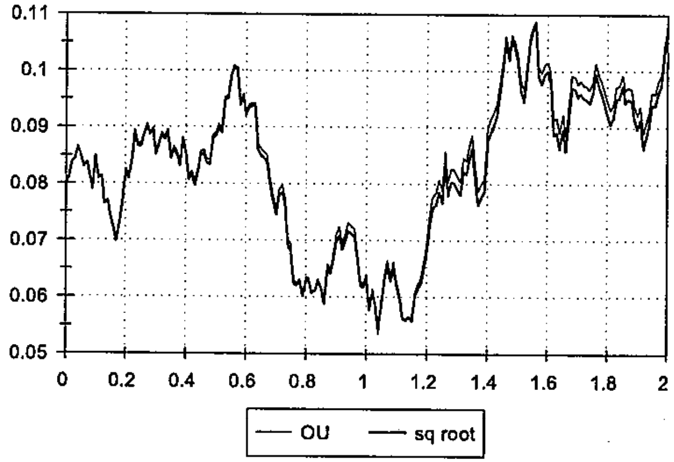
```
Public Sub Vasicek_GetBondParameters(t As Double, s As Double)
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim A As Double
    Dim B As Double
    Dim R_inf As Double
    Dim SRate As Double
    Dim SVol As Double
    r0 = VS.InitRate
    rbar = VS.LongRate
    alpha = VS.AdjSpeed
    sig = VS.Volatility
    DT = s - t

    R_inf = rbar - 0.5 * (sig / alpha) ^ 2
    A = Exp((R_inf/alpha)*(1-Exp(-alpha*DT)) - DT*R_inf - (sig*sig/(4*alpha^3))*(1-Exp(-alpha*DT))^2)
    B = (1 - Exp(-alpha * DT)) / alpha

    SRate = -Log(A) / DT + B / DT * r0
    SVol = (sig / (alpha * DT)) * (1 - Exp(-alpha * DT))

    VS.SpotRate = SRate
    VS.SpotVol = SVol
    VS.ZeroBondPrice = A * Exp(-r0 * B)
End Sub
```

(三)CIR模型的模擬

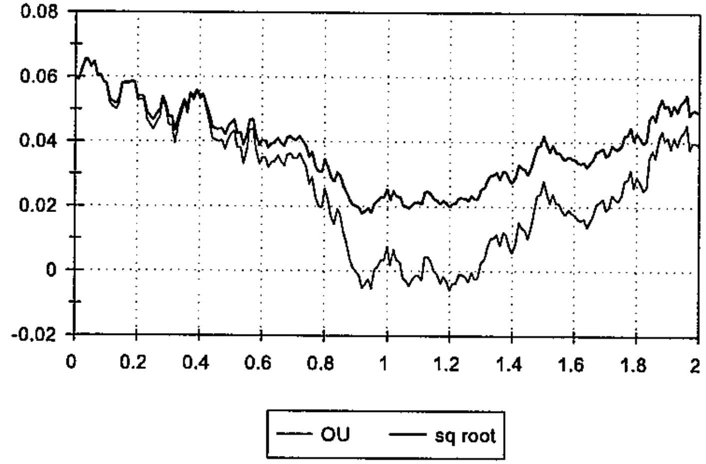
- 由(1.4)式可得下面的模擬方程式，

$$r_t = r_{t-1} + \alpha(\bar{r} - r_{t-1})(\Delta t) + \sigma\sqrt{r_t}\,\varepsilon_t\sqrt{\Delta t} \quad \text{……………………………………………………(3.2)}$$

  - 近似的模擬方法。

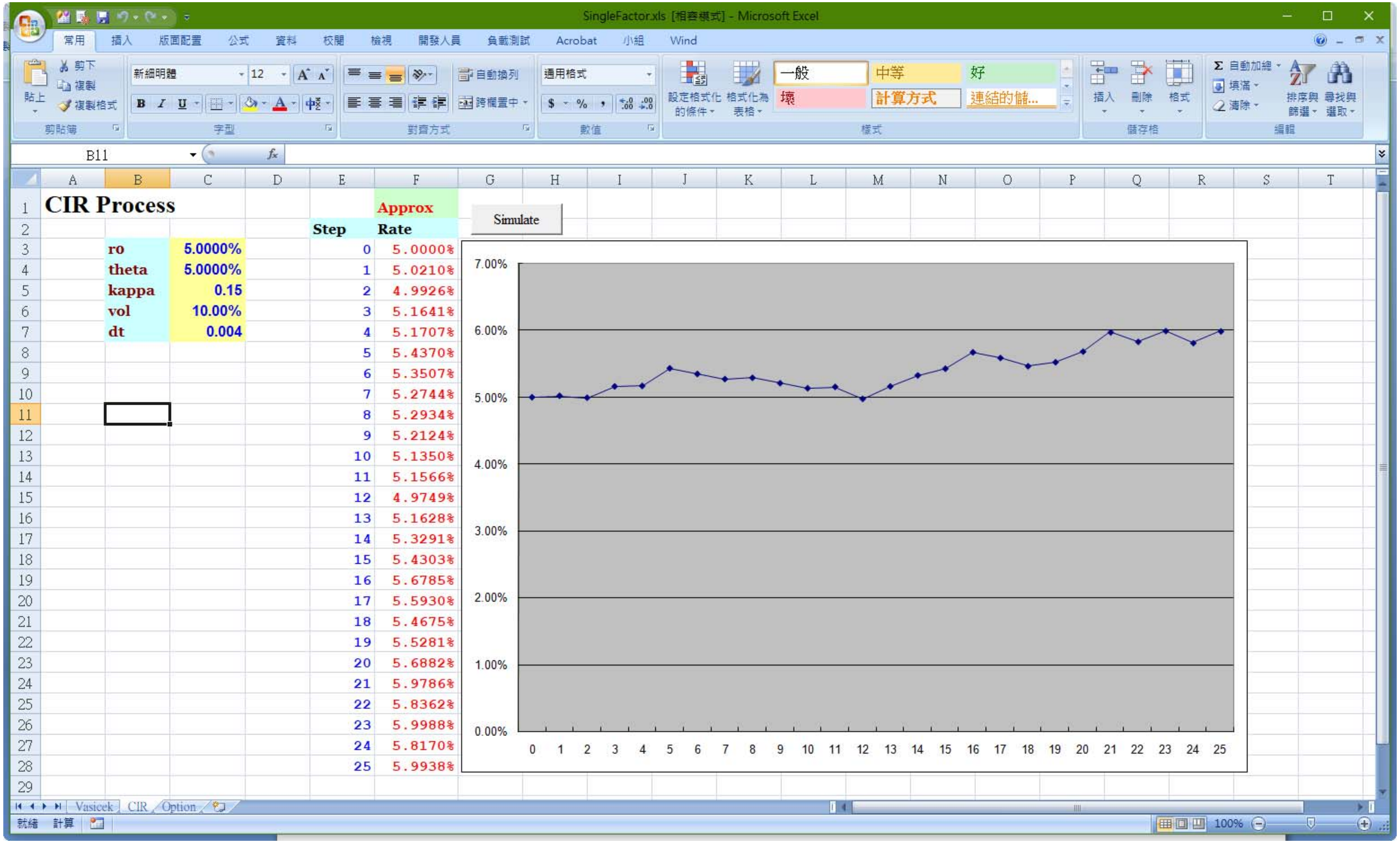


(a) Initial value $x_0 = 0.08$, same random numbers as in Figure 3.5



(b) Initial value $x_0 = 0.06$, different random numbers

Figure 3.6: A comparison of simulated paths for an Ornstein-Uhlenbeck process and a square root process. For both processes, the parameters $\theta = 0.08$ and $\kappa = \ln 2 \approx 0.69$ are used, while β is set to 0.03 for the Ornstein-Uhlenbeck process and to $0.03/\sqrt{0.08} \approx 0.1061$ for the square root process.


SingleFactor.xls [相容模式] - Microsoft Excel
CIR Process
ro 5.0000%
theta 5.0000%
kappa 0.15
vol 10.00%
dt 0.004
Approx
Step Rate
0 5.0000%
1 5.0210%
2 4.9926%
3 5.1641%
4 5.1707%
5 5.4370%
6 5.3507%
7 5.2744%
8 5.2934%
9 5.2124%
10 5.1350%
11 5.1566%
12 4.9749%
13 5.1628%
14 5.3291%
15 5.4303%
16 5.6785%
17 5.5930%
18 5.4675%
19 5.5281%
20 5.6882%
21 5.9786%
22 5.8362%
23 5.9988%
24 5.8170%
25 5.9938%
Simulate
7.00%
6.00%
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Vasicek CIR Option

```vba
Public Sub CIR_Test()
    Dim I As Integer
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim r1 As Double

    r0 = Worksheets("CIR").Range("C3").Value
    rbar = Worksheets("CIR").Range("C4").Value
    alpha = Worksheets("CIR").Range("C5").Value
    sig = Worksheets("CIR").Range("C6").Value
    DT = Worksheets("CIR").Range("C7").Value

    Call CIR_InitObj(r0, rbar, alpha, sig, DT)
    Call CIR_GetBondParameters(0, 1)
    'MsgBox CIR.ZeroBondPrice

    'Rnd (-4)
    For I = 1 To 25
        r1 = CIR_GetNextRateByApprox
        Worksheets("CIR").Cells(I + 3, 6).Value = r1
        CIR.InitRate = r1
    Next I

End Sub
```

```
Public Function CIR_GetNextRateByApprox() As Double
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim dZ As Double
    Dim r1 As Double

    r0 = CIR.InitRate
    rbar = CIR.LongRate
    alpha = CIR.AdjSpeed
    sig = CIR.Volatility
    DT = CIR.DeltaTime
    dZ = Sqr(DT) * Application.NormSInv(Rnd)

    r1 = r0 + alpha * (rbar - r0) * DT + sig * Sqr(r0) * dZ
    CIR_GetNextRateByApprox = r1
End Function
```

```
'******************* Object Module *******************************
Type CIRClass
    InitRate As Double
    LongRate As Double
    AdjSpeed As Double
    Volatility As Double
    DeltaTime As Double
    SpotRate As Double
    SpotVol As Double
    ZeroBondPrice As Double
End Type

Public CIR As CIRClass

'******************* CIR Module *******************************
Public Sub CIR_InitObj(r0, rbar, alpha, sig, DT)
    CIR.InitRate = r0
    CIR.LongRate = rbar
    CIR.AdjSpeed = alpha
    CIR.Volatility = sig
    CIR.DeltaTime = DT
    Randomize
End Sub
```

```
Public Function CIR_GetNextRateByApprox() As Double
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim dZ As Double
    Dim r1 As Double

    r0 = CIR.InitRate
    rbar = CIR.LongRate
    alpha = CIR.AdjSpeed
    sig = CIR.Volatility
    DT = CIR.DeltaTime
    dZ = Sqr(DT) * Application.NormSInv(Rnd)

    r1 = r0 + alpha * (rbar - r0) * DT + sig * Sqr(r0) * dZ
    CIR_GetNextRateByApprox = r1
End Function
```

```
Public Sub CIR_GetBondParameters(t As Double, s As Double)
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim A As Double
    Dim B As Double
    Dim phi_1 As Double, phi_2 As Double, phi_3 As Double
    Dim SRate As Double
    Dim SVol As Double

    r0 = CIR.InitRate
    rbar = CIR.LongRate
    alpha = CIR.AdjSpeed
    sig = CIR.Volatility
    DT = s - t

    phi_1 = Sqr(alpha * alpha + 2 * sig * sig)
    phi_2 = (alpha + phi_1) / 2
    phi_3 = (2 * alpha * rbar) / (sig * sig)
    A = ((phi_1 * Exp(phi_2 * DT)) / (phi_2 * (Exp(phi_1 * DT) - 1) + phi_1)) ^ phi_3
    B = (Exp(phi_1 * DT) - 1) / (phi_2 * (Exp(phi_1 * DT) - 1) + phi_1)

    SRate = -Log(A) / DT + B / DT * r0
    SVol = ((sig * Sqr(r0)) / DT) * B

    CIR.SpotRate = SRate
    CIR.SpotVol = SVol
    CIR.ZeroBondPrice = A * Exp(-r0 * B)
End Sub
```

四、利率產品的定價

(一) Vasicek模型

◆ 令短期利率的 market price of risk $\lambda(r_t,t)$ 為一常數 λ ，風險中立下的短期利率變動可表示為，

$$dr_t = \alpha(\bar{r} - r_t)\cdot dt + \sigma\cdot\left(dZ_t^Q - \lambda dt\right) \qquad (4.1)$$

$$dr_t = \alpha(\hat{r} - r_t)\cdot dt + \sigma\cdot dZ_t^Q$$

$$\hat{r} = \bar{r} - \lambda\sigma/\alpha$$

➢ dZ_t^Q 為風險中立測度下的 Weiner Process，參考(2.1)式的關係。

➢ 真實世界與風險中立測度下的利率程序有相同的數量性質。

◆ Vasicek 模型下，s 時點到期的單位面值零息債券，在 t 時點的市場價格與即期利率可求得為，

$$P(t,s)=A(t,s)\cdot Exp\left[-r\cdot B(t,s)\right] \qquad (4.2)$$

$$R(t,s)=-\frac{\ln A(t,s)}{s-t}+\frac{B(t,s)}{s-t}r$$

$$B(t,s)=\frac{1}{\alpha}\left[1-e^{-\alpha(s-t)}\right]$$

$$\ln A(t,s)=\frac{R_\infty}{\alpha}\left[1-e^{-\alpha(s-t)}\right]-(s-t)R_\infty-\frac{\sigma^2}{4\alpha^3}\left[1-e^{-\alpha(s-t)}\right]^2$$

$$R_\infty=\lim_{\tau\to\infty}R(t,\tau)=\hat{r}-\frac{\sigma^2}{2\alpha^2}$$

➢ 即期利率的波動性則為

$$\sigma_R(t,s)=\frac{\sigma}{\alpha(s-t)}\left[1-e^{-\alpha(s-t)}\right] \qquad (4.3)$$

◆ Vasicek 模型下的零息債券歐式選擇權價格可求得為，

$$c(t,T,s) = P(t,s)N(d_1) - KP(t,T)N(d_2) \quad \text{.......(4.4)}$$

$$p(t,T,s) = KP(t,T)N(-d_2) - P(t,s)N(-d_1)$$

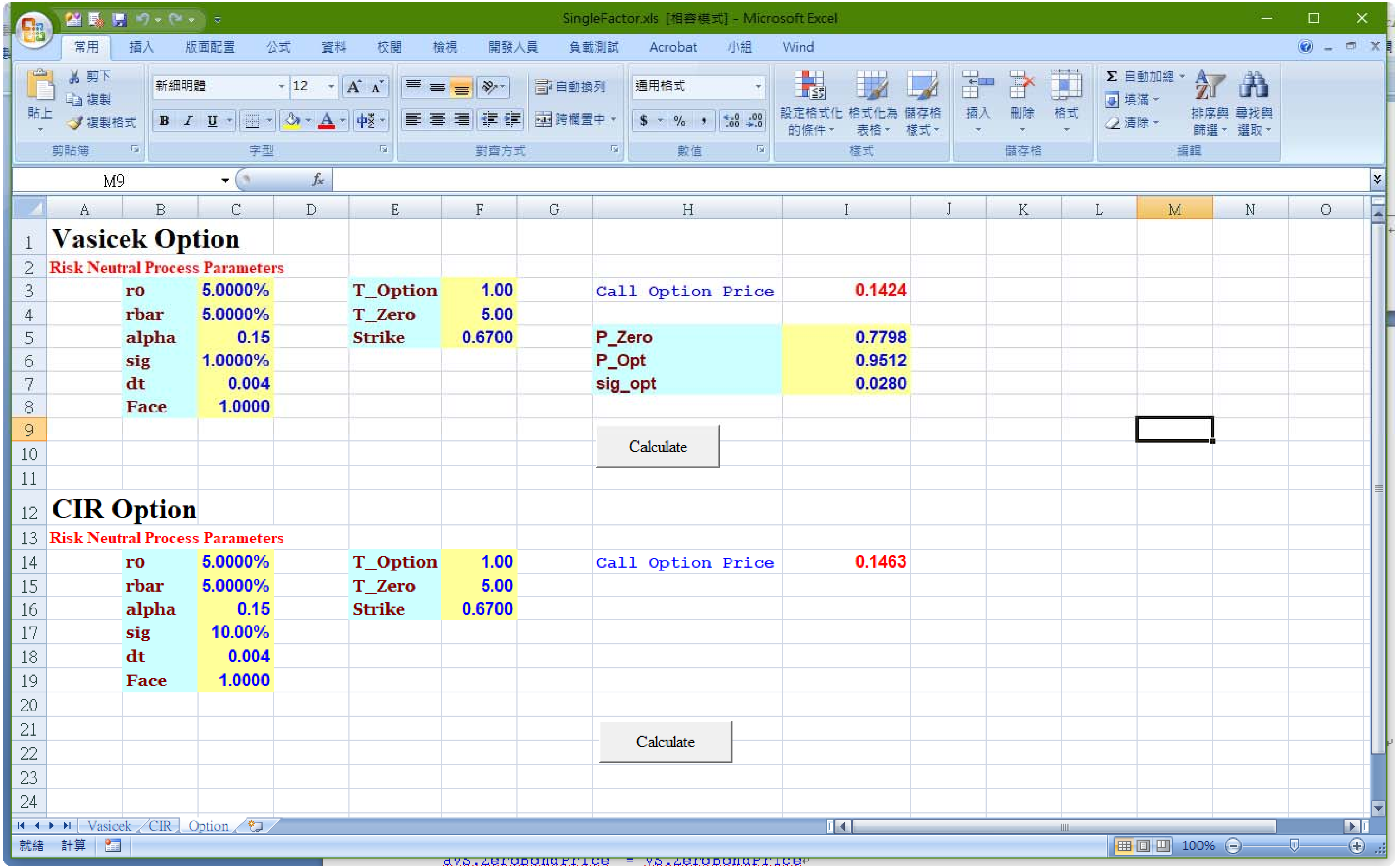
➢ 其中

$$d_1 = \frac{\ln\left(\dfrac{P(t,s)}{KP(t,T)}\right)}{\sigma_P} + \frac{\sigma_P}{2}$$

$$d_2 = d_1 - \sigma_P$$

$$\sigma_P = \frac{v(t,T)\left(1 - e^{-\alpha(s-T)}\right)}{\alpha}$$

$$v(t,T) = \sqrt{\frac{\sigma^2(1 - e^{-\alpha(s-T)})}{2\alpha}}$$


SingleFactor.xls [相容模式] - Microsoft Excel
Vasicek Option
Risk Neutral Process Parameters
ro 5.0000%
rbar 5.0000%
alpha 0.15
sig 1.0000%
dt 0.004
Face 1.0000
T_Option 1.00
T_Zero 5.00
Strike 0.6700
Call Option Price 0.1424
P_Zero 0.7798
P_Opt 0.9512
sig_opt 0.0280
Calculate
CIR Option
Risk Neutral Process Parameters
ro 5.0000%
rbar 5.0000%
alpha 0.15
sig 10.00%
dt 0.004
Face 1.0000
T_Option 1.00
T_Zero 5.00
Strike 0.6700
Call Option Price 0.1463
Calculate
Vasicek
CIR
Option

```
Public Sub Test_Vasicek_Opt()
    Dim aVS As VasicekClass
    Dim PZero As Double, POpt As Double, K As Double
    Dim TZero As Double, TOpt As Double, sig As Double
    Dim nu As Double, VSCallZeroOption As Double

    aVS.SpotRate = Worksheets("Option").Range("C3").Value
    aVS.InitRate = Worksheets("Option").Range("C3").Value
    aVS.LongRate = Worksheets("Option").Range("C4").Value
    aVS.AdjSpeed = Worksheets("Option").Range("C5").Value
    aVS.Volatility = Worksheets("Option").Range("C6").Value
    aVS.DeltaTime = Worksheets("Option").Range("C7").Value

    VS = aVS

    TOpt = Worksheets("Option").Range("F3").Value
    TZero = Worksheets("Option").Range("F4").Value
    K = Worksheets("Option").Range("F5").Value

    Call Vasicek_GetBondParameters(0, TZero)
    aVS.ZeroBondPrice = VS.ZeroBondPrice
    PZero = VS.ZeroBondPrice
    Worksheets("Option").Range("I5").Value = PZero

    Call Vasicek_GetBondParameters(0, TOpt)
    POpt = VS.ZeroBondPrice
    Worksheets("Option").Range("I6").Value = POpt

    nu = Sqr(VS.Volatility ^ 2 * (1 - Exp(-2 * VS.AdjSpeed * (TOpt))) / (2 * VS.AdjSpeed))
    sig = nu * (1 - Exp(-VS.AdjSpeed * (TZero - TOpt))) / VS.AdjSpeed
    Worksheets("Option").Range("I7").Value = sig
```

```vb
    VSCallZeroOption = Vasicek_ZeroOption("C", PZero, POpt, K, TZero, TOpt, sig)
    Worksheets("Option").Range("I3").Value = VSCallZeroOption
End Sub


'******************* Object Module ******************************
Type VasicekClass
    InitRate As Double
    LongRate As Double
    AdjSpeed As Double
    Volatility As Double
    DeltaTime As Double
    SpotRate As Double
    SpotVol As Double
    ZeroBondPrice As Double
End Type

Public VS As VasicekClass


'******************* Vasicek Module ******************************
Public Sub Vasicek_InitObj(r0, rbar, alpha, sig, DT)
    VS.InitRate = r0
    VS.LongRate = rbar
    VS.AdjSpeed = alpha
    VS.Volatility = sig
    VS.DeltaTime = DT
    Randomize
End Sub
```

```
Public Sub Vasicek_GetBondParameters(t As Double, s As Double)
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim A As Double
    Dim B As Double
    Dim R_inf As Double
    Dim SRate As Double
    Dim SVol As Double
    r0 = VS.InitRate
    rbar = VS.LongRate
    alpha = VS.AdjSpeed
    sig = VS.Volatility
    DT = s - t

    R_inf = rbar - 0.5 * (sig / alpha) ^ 2
    A = Exp((R_inf/alpha)*(1-Exp(-alpha*DT)) - DT*R_inf - (sig*sig/(4*alpha^3))*(1-Exp(-alpha*DT))^2)
    B = (1 - Exp(-alpha * DT)) / alpha

    SRate = -Log(A) / DT + B / DT * r0
    SVol = (sig / (alpha * DT)) * (1 - Exp(-alpha * DT))

    VS.SpotRate = SRate
    VS.SpotVol = SVol
    VS.ZeroBondPrice = A * Exp(-r0 * B)
End Sub
```

```
Public Function Vasicek_ZeroOption(CP_Flag As String, P_Zero As Double, P_Opt As Double, _
  K As Double, T_Zero As Double, T_Opt As Double, sig_p As Double)
    Dim d1 As Double, d2 As Double
    Dim z As Double

    d1 = Log(P_Zero / (K * P_Opt)) / sig_p + sig_p / 2
    d2 = d1 - sig_p

    z = 1#
    If CP_Flag = "P" Then
        z = -1#
    End If

    Vasicek_ZeroOption = z * (P_Zero * CND(z * d1) - K * P_Opt * CND(z * d2))
End Function
```

(二) CIR模型

- 根據此模型，短期利率的 market price of risk，$\lambda(r_t,t)$ 為

$$\lambda(r_t,t)=\frac{\lambda\sqrt{r_t}}{\sigma}$$

- 等式右側之 λ 為常數。

$$dr_t=\alpha(\bar{r}-r_t)\cdot dt+\sigma\sqrt{r_t}\left(dZ_t^Q-\lambda(r_t,t)dt\right)$$

$$dr_t=\alpha(\bar{r}-r_t)\cdot dt-\lambda r_t\cdot dt+\sigma\sqrt{r_t}dZ_t^Q=\left[\alpha\bar{r}-(\alpha+\lambda)r_t\right]dt_t+\sigma\sqrt{r_t}dZ_t^Q$$

$$dr_t=\hat{\alpha}(\hat{r}-r_t)\cdot dt+\sigma\sqrt{r_t}dZ_t^Q$$

$\hat{\alpha}=(\alpha+\lambda)$ ，$\hat{r}=\alpha\bar{r}/(\alpha+\lambda)$ 。

  - ✓ dZ_t^Q 為風險中立測度下的 Weiner Process。

- 真實世界與風險中立測度下的利率程序有相同的數量性質。

◆ CIR 模型下，s 時點到期的單位面值零息債券，在 t 時點的市場價格與即期利率可求得為，

$$P(t,s) = A(t,s)\cdot Exp\left[-r\cdot B(t,s)\right] \quad \text{......(4.5)}$$

$$R(t,s) = -\frac{\ln A(t,s)}{s-t} + \frac{B(t,s)}{s-t}r$$

$$A(t,s) = \left[\frac{\phi_1 e^{\phi_2 (s-t)}}{\phi_2\left(e^{\phi_1 (s-t)}-1\right)+\phi_1}\right]^{\phi_3}$$

$$B(t,s) = \left[\frac{e^{\phi_1 (s-t)}-1}{\phi_2\left(e^{\phi_1 (s-t)}-1\right)+\phi_1}\right]$$

$$\phi_1 \equiv \sqrt{\hat{\alpha}^2 + 2\sigma^2}\ ,\ \phi_2 \equiv \frac{(\hat{\alpha}+\phi_1)}{2}\ ,\ \phi_3 \equiv \frac{2\alpha\cdot\bar{r}}{\sigma^2}$$

$$R_\infty = \lim_{\tau\to\infty} R(t,\tau) = \frac{2\alpha\cdot\bar{r}}{\phi_1+\hat{\alpha}}$$

➢ 即期利率的波動性則為

$$\sigma_R(t,s) = \frac{\sigma\sqrt{r}}{(s-t)}B(t,s)$$

◆ CIR 模型下的零息債券歐式選擇權價格可求得為，

$$c(t,T,s)=P(t,s)\chi^2\left(2r^*[\phi+\psi+B(T,s)];\frac{4\alpha\bar{r}}{\sigma^2},\frac{2\phi^2 re^{\theta(T-t)}}{\phi+\psi+B(T,s)}\right)\text{.................................}(4.6)$$

$$-KP(t,T)\chi^2\left(2r^*[\phi+\psi];\frac{4\alpha\bar{r}}{\sigma^2},\frac{2\phi^2 re^{\theta(T-t)}}{\phi+\psi}\right)$$

- 其中

$$\theta\equiv\sqrt{\hat{\alpha}^2+2\sigma^2}\text{ ， }\phi=\frac{2\theta}{\sigma^2(e^{\theta(T-t)}-1)}$$

$$\psi=\frac{(\hat{\alpha}+\theta)}{\sigma^2}\text{ ， }r^*=\frac{\ln\left(\frac{A(T,s)}{K}\right)}{B(T,s)}$$

- A(T,s)與 B(T,s)同前定義。
- 卡方分配可以常態分配近似之。

◆ 針對卡方分配的近似估計如下，

$$\chi^2(h;f,g)\approx N(d) \qquad (4.7)$$

➢ 其中

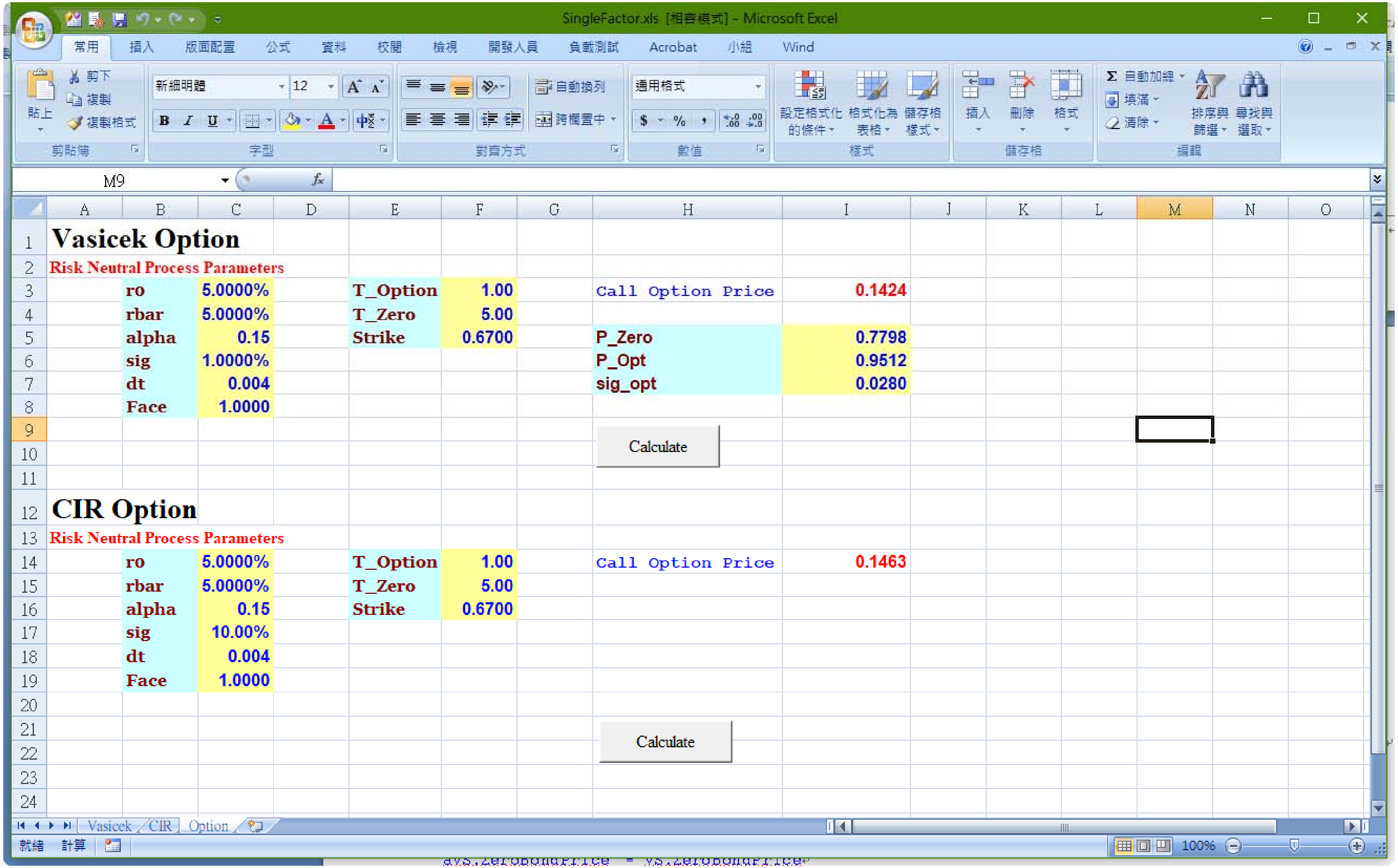
$$d=k\left(\left(\frac{h}{f+g}\right)^m-l\right)$$

$$m=1-\frac{2}{3}\frac{(f+g)(f+3g)}{(f+2g)^2}$$

$$k=\left(2m^2p[1-p(1-m)(1-3m)]\right)^{-1/2}$$

$$l=1+m(m-1)p-\frac{1}{2}m(m-1)(2-m)(1-3m)p^2$$

$$p=\frac{f+2g}{(f+g)^2}$$


SingleFactor.xls [相容模式] - Microsoft Excel
Vasicek Option
Risk Neutral Process Parameters
ro 5.0000%
rbar 5.0000%
alpha 0.15
sig 1.0000%
dt 0.004
Face 1.0000
T_Option 1.00
T_Zero 5.00
Strike 0.6700
Call Option Price 0.1424
P_Zero 0.7798
P_Opt 0.9512
sig_opt 0.0280
Calculate
CIR Option
Risk Neutral Process Parameters
ro 5.0000%
rbar 5.0000%
alpha 0.15
sig 10.00%
dt 0.004
Face 1.0000
T_Option 1.00
T_Zero 5.00
Strike 0.6700
Call Option Price 0.1463
Calculate
Vasicek
CIR
Option

```
Public Sub Test_CIR_Opt()
    Dim aCIR As CIRClass
    Dim PZero As Double, POpt As Double, K As Double
    Dim TZero As Double, TOpt As Double, sig As Double
    Dim CIRCallZeroOption As Double

    aCIR.InitRate = Worksheets("Option").Range("C14").Value
    aCIR.SpotRate = Worksheets("Option").Range("C14").Value
    aCIR.LongRate = Worksheets("Option").Range("C15").Value
    aCIR.AdjSpeed = Worksheets("Option").Range("C16").Value
    aCIR.Volatility = Worksheets("Option").Range("C17").Value
    aCIR.DeltaTime = Worksheets("Option").Range("C18").Value

    CIR = aCIR

    TOpt = Worksheets("Option").Range("F14").Value
    TZero = Worksheets("Option").Range("F15").Value
    K = Worksheets("Option").Range("F16").Value

    'Call CIR_GetBondParameters(0, TZero)
    'aCIR.ZeroBondPrice = CIR.ZeroBondPrice

    CIRCallZeroOption = CIR_ZeroOption("C", aCIR, K, TZero, TOpt)
    Worksheets("Option").Range("I14").Value = CIRCallZeroOption
End Sub
```

```
'******************* Object Module *******************************
Type CIRClass
    InitRate As Double
    LongRate As Double
    AdjSpeed As Double
    Volatility As Double
    DeltaTime As Double
    SpotRate As Double
    SpotVol As Double
    ZeroBondPrice As Double
End Type

Public CIR As CIRClass

'******************* CIR Module *******************************
Public Sub CIR_InitObj(r0, rbar, alpha, sig, DT)
    CIR.InitRate = r0
    CIR.LongRate = rbar
    CIR.AdjSpeed = alpha
    CIR.Volatility = sig
    CIR.DeltaTime = DT
    Randomize
End Sub
```

```
Public Function CIR_GetNextRateByApprox() As Double
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim dZ As Double
    Dim r1 As Double

    r0 = CIR.InitRate
    rbar = CIR.LongRate
    alpha = CIR.AdjSpeed
    sig = CIR.Volatility
    DT = CIR.DeltaTime
    dZ = Sqr(DT) * Application.NormSInv(Rnd)

    r1 = r0 + alpha * (rbar - r0) * DT + sig * Sqr(r0) * dZ
    CIR_GetNextRateByApprox = r1
End Function
```

```
Public Sub CIR_GetBondParameters(t As Double, s As Double)
    Dim r0 As Double
    Dim rbar As Double
    Dim alpha As Double
    Dim sig As Double
    Dim DT As Double
    Dim A As Double
    Dim B As Double
    Dim phi_1 As Double, phi_2 As Double, phi_3 As Double
    Dim SRate As Double
    Dim SVol As Double

    r0 = CIR.InitRate
    rbar = CIR.LongRate
    alpha = CIR.AdjSpeed
    sig = CIR.Volatility
    DT = s - t

    phi_1 = Sqr(alpha * alpha + 2 * sig * sig)
    phi_2 = (alpha + phi_1) / 2
    phi_3 = (2 * alpha * rbar) / (sig * sig)
    A = ((phi_1 * Exp(phi_2 * DT)) / (phi_2 * (Exp(phi_1 * DT) - 1) + phi_1)) ^ phi_3
    B = (Exp(phi_1 * DT) - 1) / (phi_2 * (Exp(phi_1 * DT) - 1) + phi_1)

    SRate = -Log(A) / DT + B / DT * r0
    SVol = ((sig * Sqr(r0)) / DT) * B

    CIR.SpotRate = SRate
    CIR.SpotVol = SVol
    CIR.ZeroBondPrice = A * Exp(-r0 * B)
End Sub
```

```
'*************** CIR_PDB, Price of Zeros **********************************************
'
Function CIR_PDB(r As Double, alpha As Double, rbar As Double, sigma As Double, tau As Double)
    Dim phi1 As Double, phi2 As Double, phi3 As Double

    phi1 = Sqr(alpha ^ 2 + 2 * sigma ^ 2)
    phi2 = (alpha + phi1) / 2
    phi3 = (2 * alpha * rbar) / (sigma ^ 2)

    A = ((phi1 * Exp(phi2 * tau)) / (phi2 * (Exp(phi1 * tau) - 1) + phi1)) ^ phi3
    B = ((Exp(phi1 * tau) - 1) / (phi2 * (Exp(phi1 * tau) - 1) + phi1))

    CIR_PDB = A * Exp(-B * r)
End Function ' CIR_PDB
```

```
'************** CIR_PDBO, Price of Zeros Options *************************************
'
Function CIR_PDBO(o As Integer, K As Double, T As Double, s As Double, r As Double, alpha As Double, rbar As Double,
sigma As Double)
    Dim gamma As Double, A As Double, B As Double, phi As Double
    Dim psi As Double, rs As Double, nu As Double, x1 As Double
    Dim l1 As Double, x2 As Double, l2 As Double, Q1 As Double, Q2 As Double

    gamma = Sqr(alpha * alpha + 2# * sigma * sigma)

    A = ((2*gamma*Exp((alpha+gamma)*(s-T)/2)) / ((gamma+alpha)*(Exp(gamma*(s-T))-1) + 2*gamma)) _
        ^(2 * alpha * rbar / (sigma * sigma))
    B = (2# * (Exp(gamma * (s - T)) - 1#)) / ((gamma + alpha) * (Exp(gamma * (s - T)) - 1#) + 2# * gamma)

    phi = 2# * gamma / (sigma * sigma * (Exp(gamma * T) - 1#))
    psi = (alpha + lambda + gamma) / (sigma * sigma)

    rs = Log(A / K) / B
    nu = 4# * alpha * rbar / (sigma * sigma)

    x1 = 2# * rs * (phi + psi + B)
    l1 = 2# * phi * phi * r * Exp(gamma * T) / (phi + psi + B)

    x2 = 2# * rs * (phi + psi)
    l2 = 2# * phi * phi * r * Exp(gamma * T) / (phi + psi)

    Q1 = NCSDF(x1, nu, l1)
    Q2 = NCSDF(x2, nu, l2)

    CIR_PDBO = CIR_PDB(r,alpha,rbar,sigma,s) * (1# - Q1) - K * CIR_PDB(r,alpha,rbar,sigma,T) * (1# - Q2)
End Function ' CIR_PDBO
```

```
'*********** Non-central Chi Squared Distribution Function (approximation) ****************
'
Function NCSDF(z As Double, v As Double, K As Double)
    Dim h As Double, p As Double, m As Double, x As Double

    z = z * 2
    v = v * 2
    K = K * 2
    h = 1 - 2# * (v + K) * (v + 3 * K) / (3# * (v + 2 * K) * (v + 2 * K))
    p = (v + 2 * K) / ((v + K) * (v + K))
    m = (h - 1) * (1 - 3 * h)
    x = (1# - h*p * (1# - h + 0.5*(2# - h) *m*p) - (z / (v+K))^h) / (h*Sqr((2*p*(1+m*p))))

    NCSDF = Application.NormSDist(x)
 End Function ' NCSDF
```

(三)零息債券選擇權與Caplet之關係

- 考慮一 Caplet，利率上限為 R_{cap}，期限為 t_k 到 t_{k+1}，另 R_k 為市場實際利率。則此 Caplet 在 t_{k+1} 時點之償付如下，

$$\Delta\tau \times Max\left[R_k - R_{cap}, 0\right]$$

  - 在 t_k 時點之折現值為，

$$\frac{\Delta\tau}{1 + R_k\Delta\tau} \times Max\left[R_k - R_{cap}, 0\right]$$

  - 可改寫為，

$$\left(1 + R_{cap}\Delta\tau\right) \times Max\left[\frac{1}{1 + R_{cap}\Delta\tau} - \frac{1}{1 + R_k\Delta\tau}, 0\right] \quad \text{……(4.8)}$$

- (4.8)式可看成$(1+R_{cap}\Delta\tau)$單位在 t_k 到期之歐式零息債券賣權，此債券在 t_{k+1} 到期，數量為一單位，執行價格 K_c，

$$K_c=\frac{1}{1+R_{cap}\Delta\tau}\ ,$$

$$(1+R_{cap}\Delta\tau)\times Max[\mathrm{K}_c-P(t_k,t_{k+1}),0]$$

  - (4.8)亦可改寫為，

$$Max\left[\frac{(1+R_{cap}\Delta\tau)}{1+R_{cap}\Delta\tau}-\frac{(1+R_{cap}\Delta\tau)}{1+R_k\Delta\tau},0\right]=Max\left[1-\frac{(1+R_{cap}\Delta\tau)}{1+R_k\Delta\tau},0\right]\text{......................................}(4.9)$$

    - ✓ (4.9)可看成 1 單位在 t_k 到期之歐式零息債券賣權，此債券在 t_{k+1} 到期，數量為$(1+R_{cap}\Delta\tau)$單位，執行價格 $K_c=1$。

(四)利率衍生商品視為零息債券選擇權之組合

- Jamshidian(1989)建議對於息票債券選擇權可視為零息債券選擇權之組合。
  - 在單因子模型下，我們可導出零息債券價格與其歐式選擇權之關係式。

- 一息票債券在 s_i 時點支付債息 c_i，則以期為標的之執行價格為 K，到期日為 T 之買入選擇權，$c_{CB}(t,T,\{s_i\})$，其價格可表示為

$$c_{CB}(t,T,\{s_i\}) = \sum_{i=1}^{n} c_i c(t,T,s_i,K_i)$$

  - n 表息票債券到期前支付債息的次數。
  - 第 i 筆零息債券選擇權之執行價格 K_i，由下式決定。

$$K_i = P(r^*,T,s_i)$$

  - 上式中決定債券價格的短期利率 r^*，由下式決定。

$$\sum_{i=1}^{n} c_i P(r^*,T,s_i) = K$$

- 息票債券買入選擇權可視為個別零息債券選擇權之組合，這些零息債券選擇權的到期日即為息票支付日且其執行價格需適當的調整。
  - 一息票債券在 s_i 時點支付債息 c_i，則以期為標的之執行價格為 K，到期日為 T 之賣出選擇權，$p_{CB}(t,T,\{s_i\})$，其價格可表示為

$$p_{CB}(t,T,\{s_i\}) = \sum_{i=1}^{n} c_i p(t,T,s_i,K_i)$$