

財務工程與 Excel VBA 應用

(一)選擇權定價模型

**Financial Engineering
with Excel VBA (1)
Option Pricing Model
Closed-Form**

昀騰金融科技

技術長

董夢雲 博士

目 錄

Part 1: VBA and Closed-Form Models

一、 VBA 程式環境.....	2
二、 VBA 程式設計基本.....	15
三、 條件敘述.....	27
四、 函數與程序.....	31
五、 Excel 的物件模型.....	37
六、 BS-Call 與 BS-Put.....	53
七、 Implied Volatility 與二分法尋根.....	55
八、 牛頓法尋根.....	62
九、 Exotic Option.....	65
十、 VB Source Code for Option Pricing.....	77

一、VBA 程式環境

(一)巨集簡介

➤ VBA(Visual Basic for Applications)與巨集

◆ EXCEL 2000 使用 Visual Basic for Application 6.0 (VBA 6.0)為開發工具

◆ VBA 6.0 為一整合式開發環境

☞ 包含專案總管視窗、屬性視窗、程式碼視窗

☞ 具備事件處理能力、除錯能力、與可使用 Active 控制項之表單

◆ VBA 6.0 程式設計概念與 Visual Basic 類似，OFFICE 2000 皆可運用

☞ 訊息驅動之物件導向程式設計

☞ Excel 2000、Word 2000、Access 2000、PowerPoint 2000 皆可使用

➤ 錄寫巨集的時機

➤ 錄寫巨集的過程

(二)巨集的錄製

➤ 錄製與執行巨集範例

◆ 預備錄寫巨集

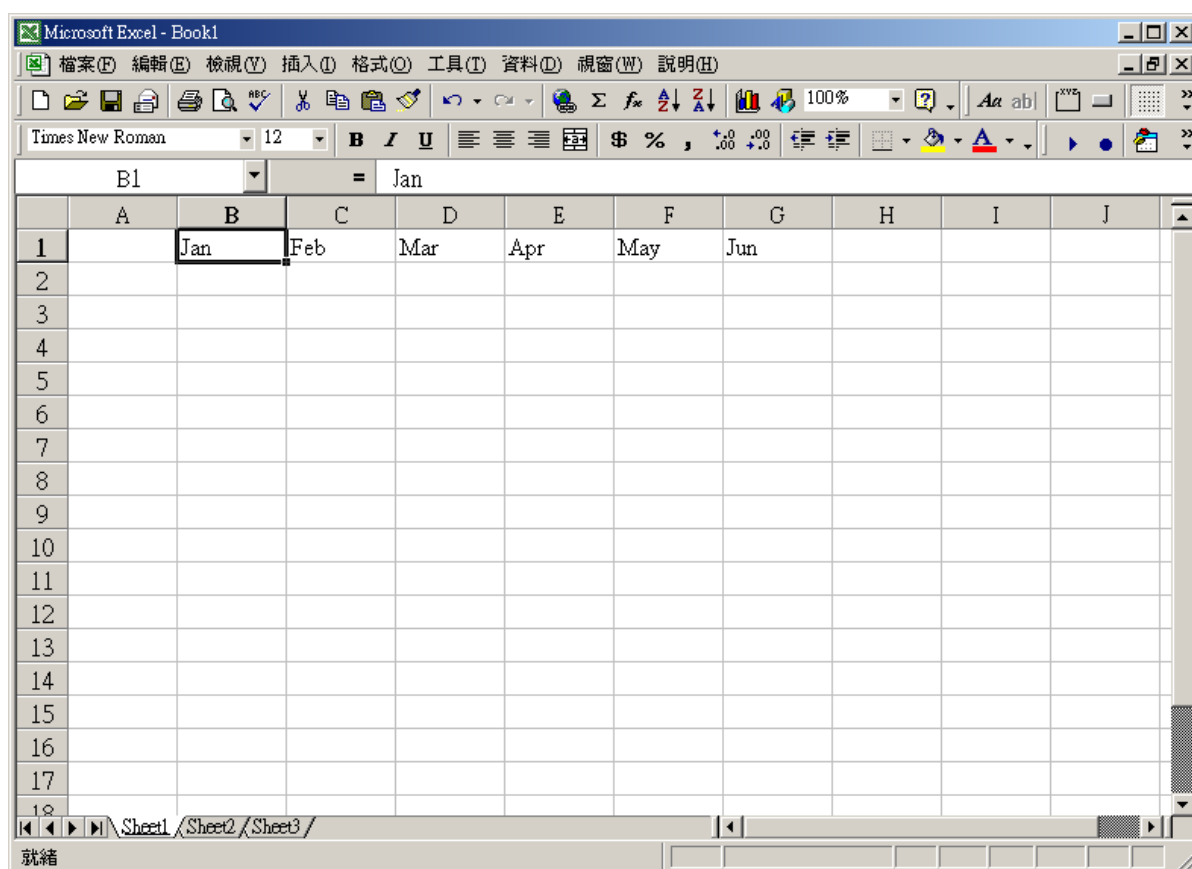
- ☞ Step1：選取＜工具／錄寫巨集／錄寫新巨集＞
- ☞ Step2：在＜巨集名稱＞方塊中輸入 **Macro1**
- ☞ Step3：在＜描述＞方塊中輸入“錄製月份標題巨集”
- ☞ Step4：在＜快速鍵＞上按 Ctrl+a
- ☞ Step5：選擇＜確定＞按鈕



◆ 開始錄寫巨集

- ☞ Step1：將游標移到 B1 儲存格
- ☞ Step2：在 B1 儲存格中輸入 Jan
- ☞ Step3：將游標移到 C1 儲存格
- ☞ Step4：在 C1 儲存格中輸入 Feb
- ☞ Step5：將游標移到 D1 儲存格
- ☞ Step6：在 D1 儲存格中輸入 Mar
- ☞ Step7：將游標移到 E1 儲存格

- ☞ Step8：在 E1 儲存格中輸入 Apr
- ☞ Step9：將游標移到 F1 儲存格
- ☞ Step10：在 F1 儲存格中輸入 May
- ☞ Step11：持續到 B1:G1 內輸入前六月份名稱
- ☞ Step12：將游標移回 B1 儲存格
- ☞ Step13：停止錄製巨集



◆ 執行巨集

- ☞ Step1：選取＜工具／巨集＞指令
- ☞ Step2：在＜巨集名稱／參照位址＞方塊中輸入或選擇 Macro1
- ☞ Step3：選擇＜執行＞按鈕

◆ 使用 Visual Basic 工具列：選取＜檢視／工具列＞之 Visual Basic 工具列，可方便巨集運用。

(三)巨集的編輯

➤ 編輯巨集

◆ 使用巨集對話方塊顯示巨集：

☞ Step1：選取＜工具／巨集＞指令

☞ Step2：在＜巨集名稱／參照位址＞方塊中輸入或選擇 **Macro1**

☞ Step3：選擇＜編輯＞按鈕

◆ 程式碼如下頁


```

Sub Macro1()
'
' Macro1 巨集表
' 錄製月份標題巨集
'
' 快速鍵: Ctrl+Shift+E
'
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "Jan"
    With ActiveCell.Characters(Start:=1, Length:=3).Font
        .Name = "Times New Roman"
        .FontStyle = "標準"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "Feb"
    With ActiveCell.Characters(Start:=1, Length:=3).Font
        .Name = "Times New Roman"
        .FontStyle = "標準"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("D1").Select
    ActiveCell.FormulaR1C1 = "Mar"

```

```

With ActiveCell.Characters(Start:=1, Length:=3).Font
    .Name = "Times New Roman"
    .FontStyle = "標準"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Range("E1").Select
ActiveCell.FormulaR1C1 = "Apr"
With ActiveCell.Characters(Start:=1, Length:=3).Font
    .Name = "Times New Roman"
    .FontStyle = "標準"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Range("F1").Select
ActiveCell.FormulaR1C1 = "May"
With ActiveCell.Characters(Start:=1, Length:=3).Font
    .Name = "Times New Roman"
    .FontStyle = "標準"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With

Range("G1").Select
ActiveCell.FormulaR1C1 = "Jun"
With ActiveCell.Characters(Start:=1, Length:=3).Font
    .Name = "Times New Roman"
    .FontStyle = "標準"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False

```

```
        .Underline = xlUnderlineStyleNone  
        .ColorIndex = xlAutomatic  
    End With  
    Range("B1").Select  
End Sub
```

```
Sub Macro1()  
  
    Range("B1").Select  
    ActiveCell.FormulaR1C1 = "Jan"  
    Range("C1").Select  
    ActiveCell.FormulaR1C1 = "Feb"  
    Range("D1").Select  
    ActiveCell.FormulaR1C1 = "Mar"  
    Range("E1").Select  
    ActiveCell.FormulaR1C1 = "Apr"  
    Range("F1").Select  
    ActiveCell.FormulaR1C1 = "May"  
    Range("G1").Select  
    ActiveCell.FormulaR1C1 = "Jun"  
    Range("B1").Select  
  
End Sub
```

(四)巨集的執行

➤ 執行巨集

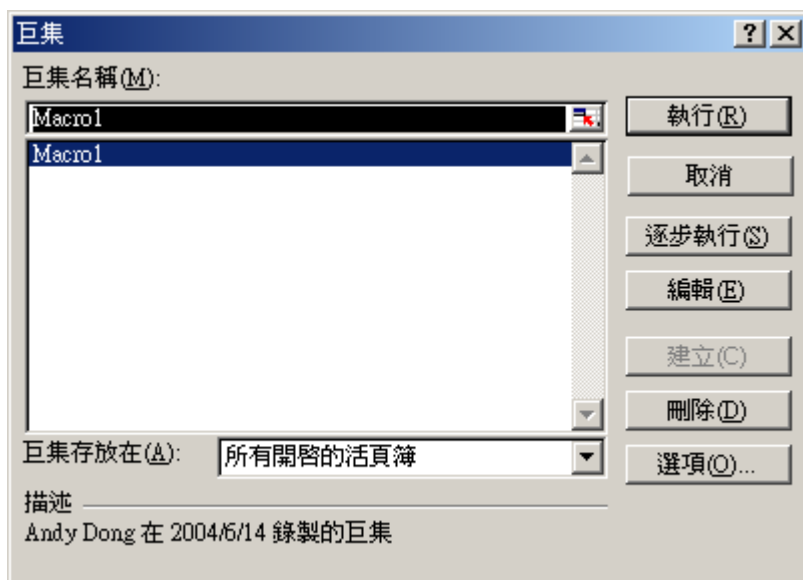
◆ 快速鍵執行：按 **Ctrl+a**

◆ 功能表執行：

☞ Step1：選取＜工具／巨集＞指令

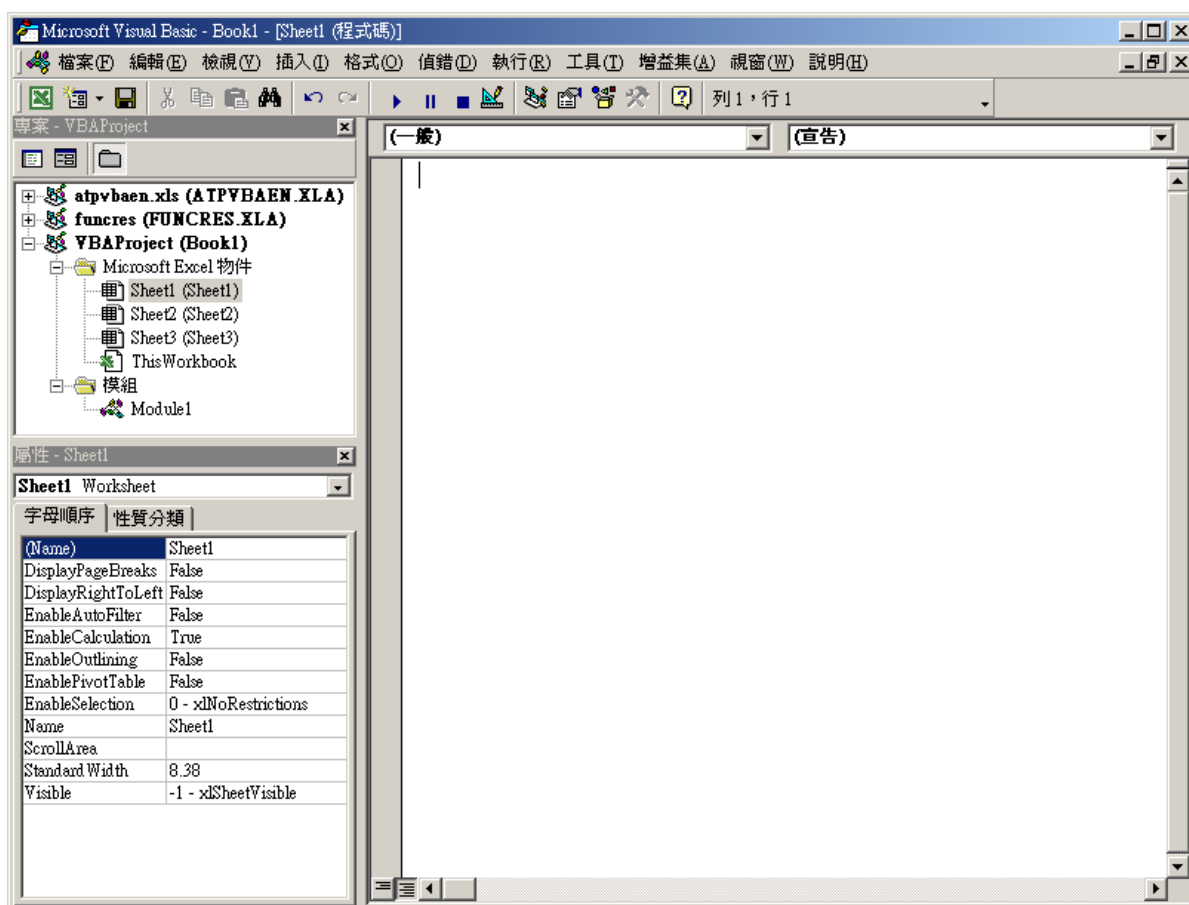
☞ Step2：在＜巨集名稱／參照位址＞方塊中輸入或選擇 **Macro1**

☞ Step3：選擇＜執行＞按鈕



(五)VBA 編輯器

- ◆ 啟動 VBE:按下 Alt+F11 進入 VBE,再按下 Alt+F11 離開 VBE 回到 Excel。
- ◆ 點選視窗工作區左方中央視窗中的 Sheet1(Sheet1)物件。出現工作區右方 Book1-Sheet1(程式碼)視窗。



◆ 專案總管

- ☞ 顯示專案內所包含的物件，主要物件有
- ☞ Microsoft Excel 物件
- ☞ 模組物件
- ☞ 表單物件

◆ 屬性視窗

- ☞ 每個物件有其屬性
- ☞ 屬性有其值

◆ 程式碼視窗

- ☞ 可撰寫程式的區域
- ☞ 視不同目的安排程式位置

二、VBA 程式設計基本

(一)VBA 之程序

◆ VBA 之程式碼稱之為”程序”，可分為兩類

- ☞ Sub 程序：由一些 VBA 陳述式組成，執行 Excel 某一件特定的工作
- ☞ 函式程序：由一些 VBA 陳述式組成，執行計算工作並可傳回一個數值

◆ 產生新 Sub 程序

- ☞ 選擇一程式碼視窗(活頁簿或模組)
- ☞ 選取<插入/程序>指令
- ☞ 鍵入名稱如 MyFirstMacro，按”確定”鈕
- ☞ 自動產生下述程式碼

```
Public Sub MyFirstMacro()
```

```
End Sub
```


◆ Sub 程序實例

```
Public Sub Example_MSG()  
    MsgBox "The area of a circle with radius 3 is"  
End Sub
```

◆ 函式程序實例

```
Public Function Area(number)  
    Area = 3.1415*number*number  
End Function
```

◆ 合並使用

```
Public Sub Example_MSG()  
    MsgBox "The area of a circle with radius 3 is " &  
        Area(3)  
End Sub
```

(二)註解

◆ 註解增加程式的可讀性

☞ 只要打上省略符號「 ' 」即可，「 ' 」後之該行文字都會被 VBA 忽略

☞ 亦可用「Rem」取代「 ' 」，Rem 只可用於一行之首

◆ 範例

```
Sub Comment()  
    ' This subroutine does nothing but comments  
    x = 0    ' x represents nothing  
    ' Display the result  
    MsgBox x  
    Rem - The whole row is remark  
End Sub
```

(三)變數之宣告

◆ 變數的用途

- ☞ 程式目的在處理資料，處理過程中資料放在電腦之中，必須有“地方”儲存這些資料，並且可隨時找到它。因此，以變數扮演此一角色。

◆ 變數的命名規則

- ☞ 首字必須為中、英文字，名稱可中英文混合，長度不可超過 255 個字元（一中文字佔二個字元）
- ☞ 名稱中不可包含空白，但可包含底線 "_"。
- ☞ 名稱不可為關鍵字、保留字。
- ☞ 英文大、小寫不分。

◆ 變數的宣告

- ☞ 變數使用前可宣告，亦可不宣告
- ☞ 宣告可增加程式執行效率
- ☞ 使用前宣告為良好習慣，可防止資料誤用

◆ 範例

```
Dim tax_rate As Single
Dim count As Integer
Dim any_obj As Variant
Dim X, Y As Double 'Y Double, X Variante
```

◆ 常數變數的意義

- ☞ 程式在處理資料的過程中，不會改變其數值的變數，如圓週率 π
 $= 3.1415926$ 。

◆ 常數變數的範例

```
Const PI As Double = 3.1415926
```

◆ 宣告的語法

- ☞ Dim 變數名稱〔資料型態宣告字元， AS 資料型態〕
- ☞ Public 變數名稱〔資料型態宣告字元， AS 資料型態〕
- ☞ Static 變數名稱〔資料型態宣告字元， AS 資料型態〕
- ☞ Private 變數名稱〔資料型態宣告字元， AS 資料型態〕
- ☞ Option Explicit：強制變數宣告

➤ 常用資料型態

◆ Integer(%)

☞ 2 Bytes, -32,768/32,767

◆ Long(&)

☞ 4 Bytes, -2,147,483,648/2,147,483,647

◆ Boolean

☞ 2 Bytes, True/False

◆ Single(!)

☞ 4 Bytes, 1.401298E-45/3.402823E38

◆ Double(#)

☞ 8 Bytes,
4.94065645841247E-324/1.79769313486232E308

◆ Date

☞ 8 Bytes, 0100/Jan/1-9999/Dec/31

◆ Currency(@)

☞ 8 Bytes, -922,337,203,685,477.5808/
922,337,203,685,477.5807

◆ Variant

☞ 16 Bytes, Same as Double

(四)運算式的表示

◆ 運算子：資料處理時，表示處理方式的符號

◆ 數值運算子

☞ 算術運算子： \wedge (指數)， $-$ (負數)， $*$ (乘)， $/$ (除)， \backslash (整數除)，
Mod(餘數)， $+$ (加)， $-$ (減)

☞ 比較運算子： $<$ ， $<=$ ， $>=$ ， $>$ ， $=$ ， $<>$

☞ 邏輯運算子：And，Or，Xor，Not

◆ 字串運算子

☞ 結合運算子： $\&$

(五)常用數學函數

◆ Log(X)

☞ 對數函數

☞ `x As Double, y As Double, y = Log(x)`

◆ Exp(x)

☞ 指數函數

☞ `x As Double, y As Double, y = Exp(x)`

◆ Sqr(x)

☞ 開平方根函數

☞ `x As Double, y As Double, y = Sqr(x)`

◆ Str(x)

☞ 轉字串函數

☞ `s As string, x As Double, s = Str(x)`

◆ Val(s)

☞ 轉數值函數

☞ `s As string, x As Double, x = Val(s)`

(六)常用輸出、入函數

➤ 輸出函數

- ◆ **MsgBox**：將訊息顯示在對話方塊中，等使用者按下按鈕，並傳回一 **Integer** 來指示使用者按下的是那一個鈕。

語法：**MsgBox(prompt[,buttons][,title][,helpfile,context])**

prompt 必要引數。做為顯示在對話方塊中之訊息的字串運算式。**prompt** 的最大長度大約是 1024 個字元，由使用字元的寬度決定。如果 **prompt** 超過一行，您可以在每一行之間用復位字元(**Chr(13)**)、換行字元(**Chr(10)**)或是復位字元與換行字元的組合(**Chr(13) & Chr(10)**)來做區隔。

buttons 選擇性引數。數值運算式，用來指出顯示按鈕的數目及形式，使用的圖示樣式，預設按鈕為何，以及訊息方塊的強制回應等。如果沒有指定，則 **buttons** 的預設值是 0。

Title 選擇性引數。顯示在對話方塊標題列中的字串運算式。如果沒有 **title**，則將應用程式的名稱放在標題列中。

Helpfile 選擇性引數。用來辨識提供給對話方塊文字感應說明的說明檔案的字串運算式。如果指定了 **helpfile**，則也必須指定 **context**。

Context 選擇性引數。數值運算式，由說明檔案的作者來指定適當的說明主題的說明主題代碼。如果指定了 **context**，則也必須指定 **helpfile**。

➤ 輸入函數

- ◆ **InputBox**：顯示一對話方塊來做為提示，等使用者輸入文字或按下按鈕，並傳回包含文字方塊內容的 **String**。

語法：**InputBox(prompt[,title][,default][,xpos][,ypos][,helpfile, context])**

- prompt** 必要引數。用來做為對話方塊訊息的字串運算式。**prompt** 的最大長度大約是 1024 個字元，由使用字元的寬度來決定。如果 **prompt** 超過一行，您可以在各行之間用復位字元(**Chr(13)**)、換行字元(**Chr(10)**)或是復位字元與換行字元的組合(**Chr(13) & Chr(10)**)來做區隔。
- title** 選擇性引數。顯示在對話方塊標題列的字串運算式。如果沒有 **title**，則以應用程式的名稱做為標題。
- default** 選擇性引數。顯示在文字方塊中的字串運算式，在沒有提供其他輸入時做為預設值。如果沒有 **default**，則文字方塊就是空白的。
- xpos** 選擇性引數。成對指定的數值運算式，用來指定對話方塊的左緣與螢幕左緣的水平距離。如果沒有 **xpos**，則對話方塊會出現在水平方向的中間。
- ypos** 選擇性引數。成對指定的數值運算式，用來指定對話方塊的上緣與螢幕的上緣的距離。如果沒有 **ypos**，對話方塊會放置於螢幕垂直 99 方向三分之一的位置。
- helpfile** 選擇性引數。字串運算式，用來指定對話方塊文字感應說明的說明檔案。如果指定了 **helpfile**，則您也必須指定 **context**。
- context** 選擇性引數。數值運算式，由說明檔案的作者指定給某個說明主題的說明主題代碼。如果指定了 **context**，則也必須指定 **helpfile**。

(七)Normal PDF*

➤ 標準常態分配之機率密度函數

$$\text{NorPDF}(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

```
Public Sub NorPdf()  
    Const Pi As Double = 3.1415926  
    Dim ans As Double  
  
    x = 0  
    ans = (Exp(-x * x / 2)) / ((2 * Pi) ^ 0.5)  
    MsgBox ans  
End Sub  
  
Public Function NorPdf(x As Double) As Double  
    Const Pi As Double = 3.1415926  
  
    NorPdf = (Exp(-x * x / 2)) / ((2 * Pi) ^ 0.5)  
End Function
```

三、條件敘述

(一)If ... Then ... [End If]

```
If condition Then true_instructions  
    [Else false_instructions]  
End If
```

```
Sub GreetMe()  
    If Time < 0.5 Then MsgBox "Good Morning"  
End Sub
```

```
Sub GreetMe2()  
    If Time < 0.5 Then MsgBox "Good Morning"  
    If Time >= 0.5 Then MsgBox "Good Afternoon"  
End Sub
```

```
Sub GreetMe3()  
    If Time < 0.5 Then  
        MsgBox "Good Morning"  
    Else  
        MsgBox "Good Afternoon"  
    End If  
End Sub
```

(二)Do ... Loop

```
Do While [condition]
```

```
    [instructions]
```

```
Loop
```

```
Do
```

```
    [instructions]
```

```
Loop While [condition]
```

```
Sub DoubleValue()
```

```
    Range("A1").Select
```

```
    Do While not IsEmpty(ActiveCell)
```

```
        ActiveCell.Value = 2 * ActiveCell.Value
```

```
        ActiveCell.Offset(1,0).Select 'next row
```

```
    Loop
```

```
End Sub
```

(三)Normal CDF*

➤ 標準常態分配之累積機率密度函數

$$N(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du$$

➤ $N(x)$ 之近似多項式：

$$N(x) = 1 - (a_1 k + a_2 k^2 + a_3 k^3) \bullet N'(x) \quad x \geq 0$$

$$N(x) = 1 - N(-x) \quad x < 0$$

$$k = \frac{1}{1 + \alpha x}$$

$$\alpha = 0.33267$$

$$a_1 = 0.4361836$$

$$a_2 = -0.1201676$$

$$a_3 = 0.9372980$$

$$N'(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

◆ 函數型式

```
Public Function NorCdf(x As Double) As Double
    Dim alph As Double, k As Double
    Dim a1 As Double, a2 As Double, a3 As Double
    Dim ans As Double

    alpha = 0.33267
    a1 = 0.4361836
    a2 = -0.1201676
    a3 = 0.937298
    k = 1 / (1 + alpha * x)
    If x >= 0 Then
        ans = 1 - (a1 * k + a2 * k * k + a3 * k * k * k) *
            NorPdf(x)
    Else
        ans = 1 - NorCdf(-x)
    End If
    NorCdf = ans
End Function
```

◆ 呼叫副程式

```
Public Sub CDFCall()
    Dim x As Double
    x = Val(InputBox("Input the ending point of Normal
        CDF"))
    MsgBox "The CDF of Normal Distribution at "&Str(x)&"
        = "&Str(NorCdf(x))
End Sub
```

四、函數與程序

(一)函數格式

➤ Sub

```
[Private|Public][Static]Sub name[(Arglist)]  
    [instructions]  
    [Exit Sub]  
    [instructions]  
End Sub
```

- ◆ **Private**：只有在宣告的模組才可使用
- ◆ **Public**：活頁簿所有的模組都可使用
- ◆ **Static**：此程序的變數在程序結束時被保存起來
- ◆ **Arglist**：負責接受被傳到程序的參數
- ◆ **Exit Sub**：強制立刻離開程序

<<BS Formula Exercise>>

➤ Function

```
[Private|Public][Static]Function name[(Arglist)][As  
Type]  
    [instructions]  
    [name = expression]  
End Function
```


(二)呼叫函式

- ◆ 輸入程序的名稱，後面接它的參數，以逗號分隔

```
MsgBox "Hello world!"
```

- ◆ 使用 Call 關鍵字，後接程序的名稱，再接它的參數，並將之放在括號中以逗號分隔

```
Call MsgBox("Hello world!")
```

- ◆ 不同模組間呼叫，要加上模組名稱於程序的名稱之前

```
MyModule.MySub
```

(三)傳遞參數

➤ 傳址呼叫(Default)

```
Sub Main()  
    MyValue = 10  
    Call Process(MyValue)  
    MsgBox MyValue    '100  
End Sub
```

```
Sub Process(ByRef TheValue As Double)  
    TheValue = TheValue * 10  
    MsgBox TheValue    '100  
End Sub
```

➤ 傳值呼叫

```
Sub Main()  
    MyValue = 10  
    Call Process(MyValue)  
    MsgBox MyValue    '10  
End Sub
```

```
Sub Process(ByVal TheValue As Double)  
    TheValue = TheValue * 10  
    MsgBox TheValue    '100  
End Sub
```

(四)Excel 函數呼叫

◆ 引用函數的原則

- ☞ 優先使用 VBA 函數
- ☞ 其次使用 Excel 工作表函數
- ☞ 最後自建函數

◆ 查詢 Excel 工作表函數

- ☞ 在 VBE 中按 <F2>，顯示瀏覽物件對話方塊

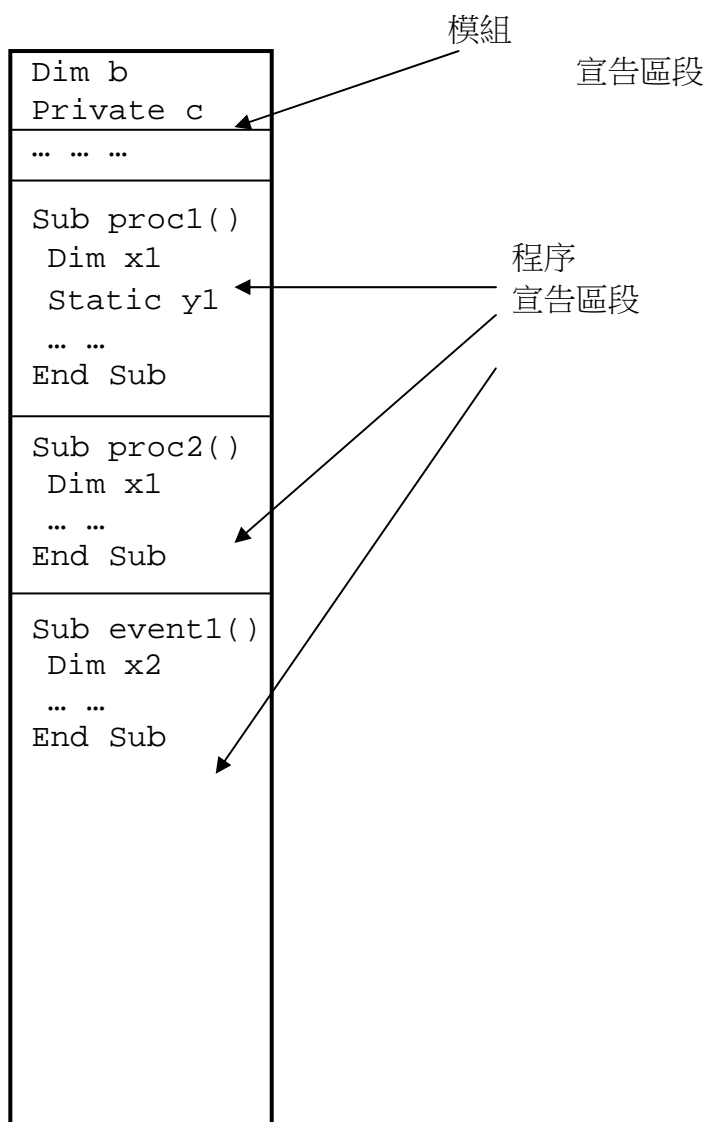
◆ 工作表函數

- ☞ Application.WorkSheetFunction.FcnName() or
- ☞ Application.FcnName()

```
Sub ShowMax( )  
    TheMax = Application.WorkSheetFunction.Max _  
        (Range( "A1:A50" ) )  
    MsgBox TheMax  
End Sub
```

(五)變數的可見性

- ◆ 區域（程序）變數：Dim，Static
- ◆ 模組變數：Dim，Private
- ◆ 共用變數：Public



五、Excel 的物件模型

(一)物件概論

- ◆ 電腦系統與物件
- ◆ 電腦系統目的在解決現實世界的問題
- ◆ 現實世界的問題起源於物與物的交互關聯（互動）
- ◆ 電腦系統必須實作出”物件”與”互動”
- ◆ 物件由”屬性”表徵，物件的行為用”方法”完成
- ◆ 互動的行為由”訊息”觸發”事件”

- ◆ 圖書管理系統為例
 - ❧ 物件有讀者、圖書與借書
 - ❧ 讀者物件之屬性有姓名、身份字號、可借書數與已借書數
 - ❧ 借書物件之屬性有借書序號、圖書號碼、借書日期與到期日
 - ❧ 圖書物件之屬性有圖書號碼、書名、目前狀態
 - ❧ 讀者物件之方法有借書、還書與預約
 - ❧ 圖書物件之方法有設定狀態與讀取狀態
 - ❧ 借書物件之方法有成立借書、刪除借書與重設到期日

◆ 可以下圖模型表達

◆ 專案總管之專案物件有

- ☞ Microsoft Excel 物件與表單物件
- ☞ 各物件之屬性列於屬性視窗
- ☞ 各物件之方法撰寫於程式碼檔
- ☞ 物件具有層次結構
- ☞ 表單物件上有按鈕物件與標籤物件

◆ 屬性視窗

- ☞ 二欄式清單
- ☞ 第一欄為屬性名稱
- ☞ 第一欄為屬性設定值

◆ 專案屬性：VBA 編輯環境下<工具/VBA Project 屬性>

(二)Excel 物件之屬性與方法

➤ Excel 物件概念

- ◆ Excel 本身即為一物件，它包含著其他的物件。一個 Excel 應用程式物件包含下面物件

- 🔗 Addin
- 🔗 CommandBar
- 🔗 VBE
- 🔗 WorkBook
- 🔗 WorkSheetFunction

- ◆ WorkBook 物件包含下面物件

- 🔗 Chart
- 🔗 CommandBar
- 🔗 VBProject
- 🔗 WorkSheet

◆ WorkSheet 物件包含下面物件

🔗 ChartObject

🔗 PageSetup

🔗 QueryTable

🔗 Range

🔗 Shape

➤ 物件的參照(Reference)

- ◆ VBA 的集合物件：一堆相同型態物件的集合，常用集合物件有：

- ☞ Workbooks 集合物件
- ☞ Worksheets 集合物件
- ☞ Charts 集合物件
- ☞ Sheets 集合物件
- ☞ Shapes 集合物件

- ◆ 參照物件：透過物件的從屬關係，指涉出欲處理的資料，此即物件參照

- ☞ `Application.Workbooks("Book1").Worksheets("Sheet1").Range("A1")`，完整參照
- ☞ `Workbooks("Book1").Worksheets("Sheet1").Range("A1")`，開起數個活頁簿之參照
- ☞ `Worksheets("Sheet1").Range("A1")`，開起數個工作表之參照，Book1 為工作中活頁簿
- ☞ `Range("A1")`，Sheet1 為工作中工作表，`Range()` 為範圍物件

➤ 物件的屬性

◆ Value 屬性：單一儲存格的範圍物件

```
Sub ChangeValue()  
    Dim Content As Single  
    Content = InputBox("Enter a Value")  
    Worksheets("Sheet1").Range("A1").Value = Content  
End Sub
```

```
Sub ShowValue()  
    Dim Content As Single  
    Content = Worksheets("Sheet1").Range("A1").Value  
    MsgBox Content  
End Sub
```

```
Sub SetValue()  
    Call ChangeValue  
    Call ShowValue  
End Sub
```

◆ 部份屬性為唯讀

```
Sub CountBooks()  
    MsgBox Workbooks.Count  
End Sub
```

➤ 物件的方法

◆ 範圍物件之方法

🌀 計算 A1 格之公式

```
Sub CalcCell()  
    Worksheets("Sheet1").Range("A1").Calculate  
End Sub
```

🌀 將 A1 內容拷貝至 B1

```
Sub CopyOne()  
    Worksheets("Sheet1").Activate  
    Range("A1").Copy Range("B1")  
End Sub
```

◆ 集合物件之方法

🌀 新增一活頁簿

```
Sub AddAWorkbook()  
    Workbooks.Add()  
End Sub
```

➤ 使用範圍物件

◆ Range

- 🔗 `Range("A1:A5")`
- 🔗 `Range("PriceList")`
- 🔗 整列：`Range("3:3")`，整行：`Range("D:D")`

◆ Cells

- 🔗 C2 儲存格：`Cells(2,3)`
- 🔗 `Range(Cells(1,1), Cells(10,10))=Range("A1:J10")`
- 🔗 `Range("A1:J10").Value = 99`

◆ Offset

- 🔗 C2 儲存格：`Range("A1").Offset(1,2)`
- 🔗 A1 儲存格：`Range("C2").Offset(-1,-2)`

◆ Rows

- 🔗 `Rows("1:5")`

◆ Columns

- 🔗 `Columns("A:C")`

➤ 常用範圍物件屬性

◆ Value

- ☞ `MsgBox Worksheets("Sheet1").Range("A1").Value`
- ☞ `Worksheets("Sheet1").Range("A1:C3").Value = 75`

◆ Text : Read Only

- ☞ 將儲存格之值以格式化方式顯示
- ☞ `MsgBox Worksheets("Sheet1").Range("A1").Text`

◆ Count : Read Only

- ☞ 顯示儲存格之數目
- ☞ `MsgBox Range("A1:C3").Count (9)`

◆ Column & Row : RO

- ☞ 傳回儲存格所在之行號或列號
- ☞ `MsgBox Sheets("Sheet1").Range("F3").Row (3)`

◆ Address : RO

- ☞ 顯示儲存格位址
- ☞ `MsgBox Range(Cells(1,1), Cells(5,5)).Address (A1:E5)`

◆ HasFormula : RO

- ☞ 儲存格內是否有定義公式
- ☞ `MsgBox Range("A1").HasFormula`

◆ Font

☞ `Range("A1").Font.Bold = True`

◆ Formula

☞ 設定一儲存格內的公式

☞ `Range("A13").Formula = "=Sqrt(100)"`

◆ NumberFormat

☞ 以字串方式顯示數值格式

☞ `Columns("A:A").NumberFormat = "0.00%"`

➤ 常用範圍物件方法

◆ Select

☞ 選取一個範圍物件，要先 Activate

```
Sheets("Sheet1").Activate  
Range("A1:C12").Select
```

◆ Copy & Paste

```
Sub CopyRange()  
    Range("A1:A12").Select  
    Selection.Copy  
    Range("C1").Select  
    ActiveSheet.Paste  
End Sub
```

```
Sub CopyRange2()  
    Range("A1:A12").Copy Range("C1")  
End Sub
```

◆ Clear

☞ Columns("D:D").Clear

☞ ClearContents, ClearFormats 方法

◆ Delete

☞ Row("6:6").Delete

(三) Excel 物件之事件與訊息

◆ 物件之行為由訊息通知，構成事件的發生

- ☞ 訊息由作業系統產生，通知程式
- ☞ 訊息範例如：活頁簿打開，按鈕被壓，滑鼠通過
- ☞ 事件帶出物件的方法

◆ 程式碼視窗分為不同事件處理

- ☞ Worksheet：工作表事件
- ☞ Workbook：活頁簿事件
- ☞ Chart：圖表事件
- ☞ Userform：表單與其上控制項事件
- ☞ 一般：變數宣告，通用副程式之處理

➤ 物件的事件

◆ Excel 中切換到不同活頁簿時，一個 Activate「事件」即會發生。

◆ 常用事件：

```
Private Sub Worksheet_Activate()
```

```
End Sub
```

```
Private Sub Workbook_Activate()
```

```
End Sub
```

```
Private Sub Workbook_Open()
```

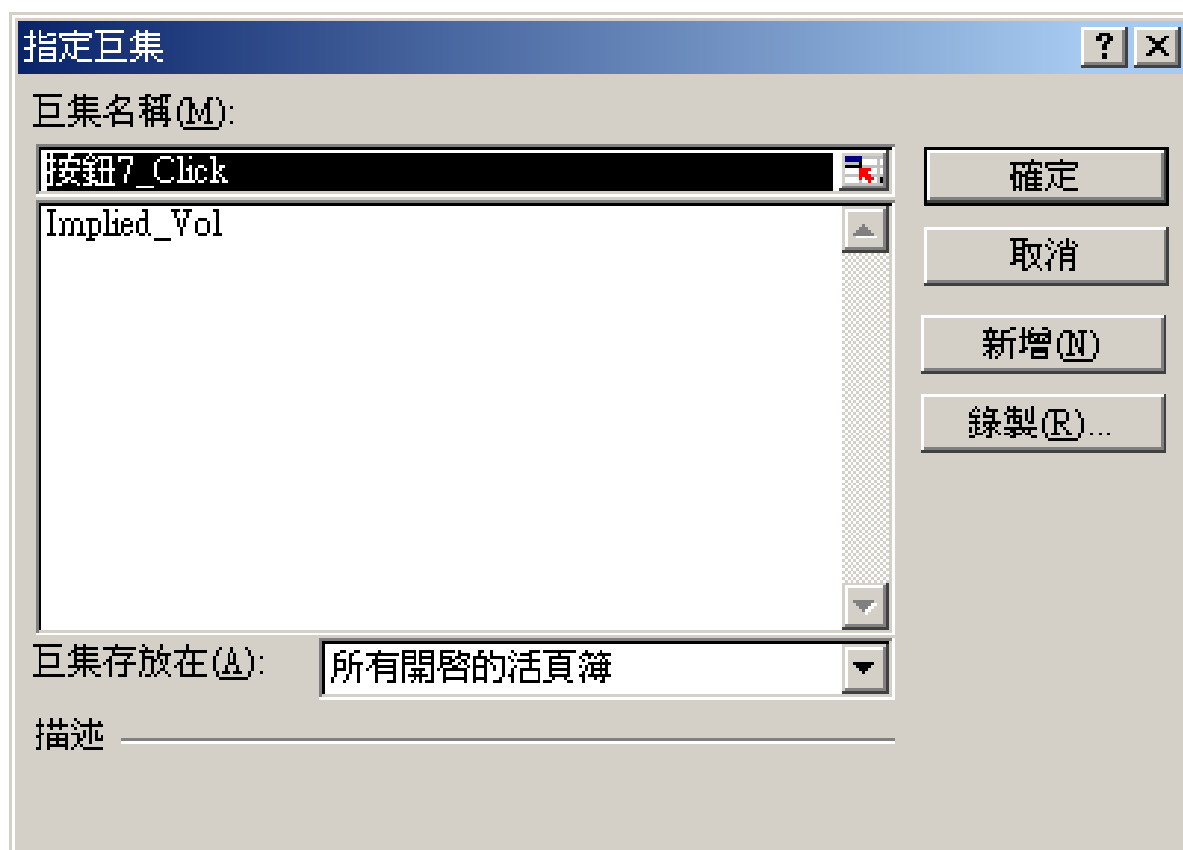
```
End Sub
```

➤ Button 的事件

◆ 在試算表頁面上，由表單中將 Button 拖曳下來，指定巨集的對話框自動出現。

☞ 選取特定之巨集，將按鈕事件與巨集的執行連結起來。

☞ 爾後，只要按 Button 便會執行該巨集。



六、BS-Call 與 BS-Put*

(一)選擇權定價公式的應用

$$C = Se^{-r_f T} N(d_1) - Ke^{-rT} N(d_2)$$

$$P = Ke^{-rT} N(-d_2) - Se^{-r_f T} N(-d_1)$$

◆ GK 定價公式程式碼

```
Public Sub Op_Cvalue(S As Double, K As Double, r As Double,
    f As Double, sd As Double, T As Double, Price As
    Double)
```

```
    Dim d1 As Double, d2 As Double
    Dim Class As String, Position As String
```

```
    d1 = (Log(S/K)+(r-f+sd*sd/2)*T)/(sd*Sqr(T))
    d2 = d1-sd*Sqr(T)
```

```
    Price = S*Exp(-f*T)*NorCDF(d1)-K*Exp(-r*T)*
        NorCDF(d2)
```

```
End Sub
```

```
Public Sub Op_Pvalue(S As Double, K As Double, r As Double,
    f As Double, sd As Double, T As Double, Price As
    Double)
```

```
    Dim d1 As Double, d2 As Double
    Dim Class As String, Position As String
```

```
    d1 = (Log(S/K)+(r-f+sd*sd/2)*T)/(sd*Sqr(T))
    d2 = d1-sd*Sqr(T)
```

```
    Price = K*Exp(-r*T)*NorCDF(-d2)-S*Exp(-f*T)*
        NorCDF(-d1)
```

```
End Sub
```

```

Public Function NorCDF(d As Double) As Double
    Dim ans As Double, g As Double
    Const a1 = 0.4361836
    Const a2 = -0.1201676
    Const a3 = 0.937298

    g = 1 / (1 + 0.33267 * d)

    If d > 0 Then
        ans = 1-(a1*g+a2*g*g+a3*g*g*g)* NorPDF(d)
    Else
        ans = 1 - NorCDF(-d)
    End If

    NorCDF = ans
End Function

```

```

Public Function NorPDF(x As Double) As Double
    NorPDF = Exp(-x * x / 2) / Sqr(2 * pi)
End Function

```

<<WorkSheet Exercise>>

<<Greeks Exercise>>

七、Implied Volatility 與二分法尋根*

(一)Implied Volatility

- ◆ 波動性，Volatility，乃衡量資產價格變動的程度。
- ◆ 為何需要知道波動性
 - ☞ 權利型式的金融工具，其價值受到波動性的影響
 - ☞ Example: Stock Option, Interest Rate Option, Convertible Bond
 - ☞ 計算 Greeks 與風險值時，需要波動性
- ◆ 問題：大部份金融資產價格的波動性無法直接由市場觀察到
 - ☞ OTC Products：直接報 Vol
 - ☞ Listed Products：報價格

➤ 歷史波動性

◆ 使用過去之歷史資料求得資產價格的波動性

☞ 波動性與採樣時間週期的平方根成正比

◆ 估計步驟

☞ 取得歷史價格資料 $P_0, P_1, P_2, \dots, P_N$ 。

☞ 計算報酬資料 $R_1, R_2, R_3, \dots, R_N$ 。

$$R_n = \ln \left[\frac{P_n}{P_{n-1}} \right]$$

☞ 求取報酬之標準差

☞ 轉換為年標準差

➤ 隱含波動性

◆ 有限的來源與產品

☞ Market Vol 可由一些資訊源取得，如 Reuters、Bloomberg。

☞ 只限於特定的產品。

◆ 資料的透明性

☞ 只有交易所交易的產品價格有較佳可靠性。

◆ 模型的缺點

☞ Black-Scholes 模型有太多的假設。

☞ 隱含波動性只適用一特定期限，無法內、外差。

$$C = f(S, K, T, r_d, r_f, \sigma)$$

$$\sigma = f^{-1}(C, S, K, T, r_d, r_f)$$

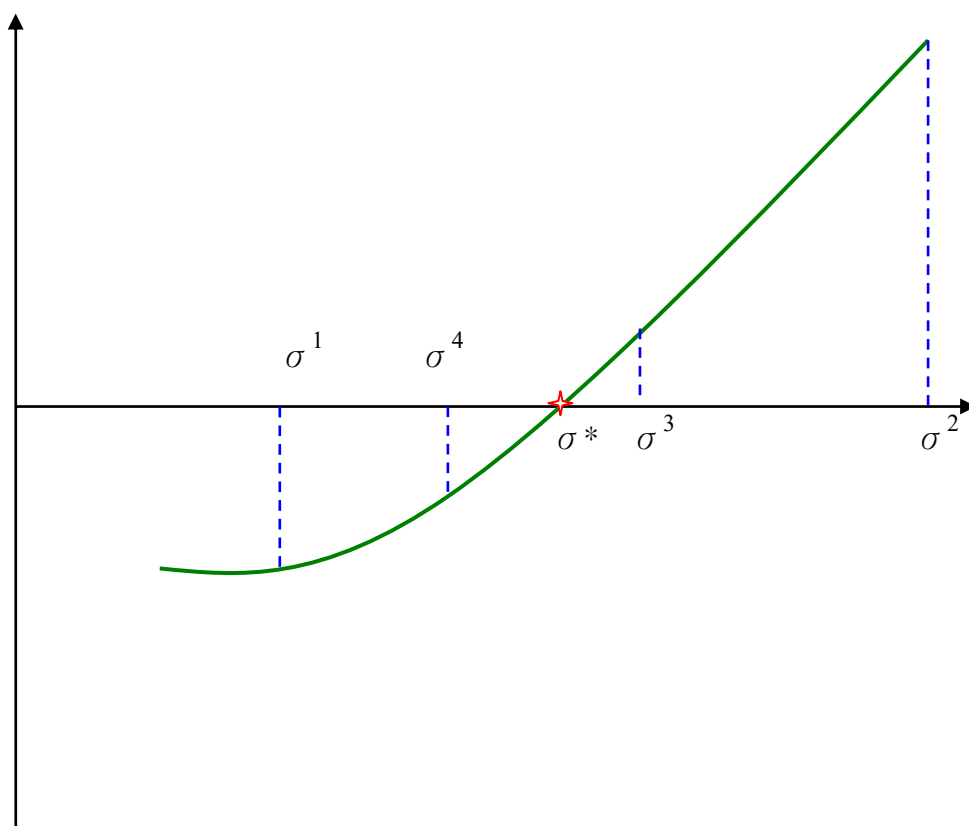
(二) Bisection Method

- ◆ 已知 C^* ，要求 σ^* ，如下式

$$C^* = f(S, K, T, r_d, r_f, \sigma^*)$$

- ◆ 轉化為下式求根

$$g(\sigma^*) = f(S, K, T, r_d, r_f, \sigma^*) - C^* = 0$$



➤ Pseudo Code

Input: ε , a , and b ($b > a$ and $g(a) \cdot g(b) < 0$);

Real: $length$, c ;

$Length := b - a$;

While [$length > \varepsilon$]

{

$c := (a + b) / 2$

 if [$g(c) = 0$] return c ;

 else if [$g(a)g(c) < 0$] $b := c$;

 else $a := c$;

$length := b - a$;

}

return c ;

(三)Code for BS Options

```
Const pi = 3.1415926
Public Sub Implied_Vol()
    Dim S As Double, K As Double, r As Double, f As Double
    Dim sd1 As Double, sd2 As Double, T As Double
    Dim P As Double, std As Double, err As Double
    Dim tol As Double, value As Double
    S = Worksheets("Implied Vol").Range("A4").value
    K = Worksheets("Implied Vol").Range("B4").value
    r = Worksheets("Implied Vol").Range("E4").value
    f = Worksheets("Implied Vol").Range("F4").value
    T = Worksheets("Implied Vol").Range("D4").value
    P = Worksheets("Implied Vol").Range("B7").value
    tol = 0.0001
    sd1 = 0.0001
    sd2 = 2
    std = (sd1 + sd2) / 2
    Call Op_Cvalue(S, K, r, f, std, T, value)
    err = Abs(value - P)

    Do While (err >= tol)
        If (value >= P) Then sd2 = std
        If (value < P) Then sd1 = std
        std = (sd1 + sd2) / 2
        Call Op_Cvalue(S, K, r, f, std, T, value)
        err = Abs(value - P)
    Loop

    Worksheets("Implied Vol").Range("B9").value = std
End Sub

Public Sub Op_Cvalue(S As Double, K As Double, r As Double,
    f As Double, sd As Double, T As Double, Price As Double)
    Dim d1 As Double, d2 As Double
    Dim Class As String, Position As String

    d1 = (Log(S / K) + (r - f + sd * sd / 2) * T) / (sd * Sqr(T))
    d2 = d1 - sd * Sqr(T)
    Price = S * Exp(-f * T) * NorCDF(d1) - K * Exp(-r * T) * NorCDF(d2)
End Sub

Public Sub Op_Pvalue(S As Double, K As Double, r As Double, f As
    Double, sd As Double, T As Double, Price As Double)
    Dim d1 As Double, d2 As Double
    Dim Class As String, Position As String

    d1 = (Log(S / K) + (r - f + sd * sd / 2) * T) / (sd * Sqr(T))
    d2 = d1 - sd * Sqr(T)
    Price = K * Exp(-r * T) * NorCDF(-d2) - S * Exp(-f * T) * NorCDF(-d1)
End Sub
```

```

Public Function NorCDF(d As Double) As Double
    Dim ans As Double, g As Double
    Const a1 = 0.4361836
    Const a2 = -0.1201676
    Const a3 = 0.937298

    g = 1 / (1 + 0.33267 * d)
    If d >= 0 Then
        ans = 1 - (a1 * g + a2 * g * g + a3 * g * g * g) * NorPDF(d)
    Else
        ans = 1 - NorCDF(-d)
    End If

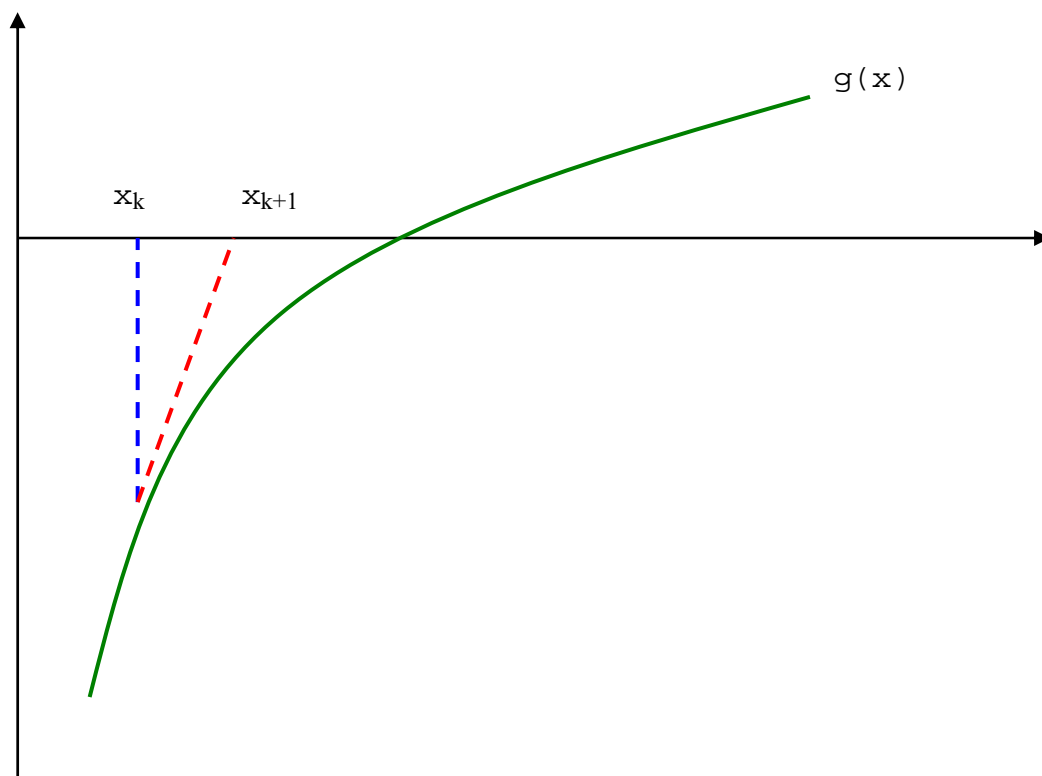
    NorCDF = ans
End Function

Public Function NorPDF(x As Double) As Double
    NorPDF = Exp(-x * x / 2) / Sqr(2 * pi)
End Function

```

八、牛頓法尋根*

(一)Newton-Raphson Method



$$g(\sigma^*) = f(S, K, T, r_d, r_f, \sigma^*) - C^* = 0$$

$$g(x_k) = -(x_{k+1} - x_k) \times g'(x_k)$$

$$x_{k+1} \equiv x_k - \frac{g(x_k)}{g'(x_k)} \quad \text{Price_error / Vega}$$

$$g'(x) = \frac{\partial g}{\partial x} = \frac{\partial f}{\partial \sigma} = \text{vega} = \frac{\partial C}{\partial \sigma}$$

$$\frac{\partial C}{\partial \sigma} = S\sqrt{T}e^{-yT}\Phi(d_1)$$

➤ Pseudo Code

Input: ε , \mathbf{x}_{init} ;

Real: \mathbf{x}_{new} , \mathbf{x}_{old} ;

$\mathbf{x}_{\text{old}} := \mathbf{x}_{\text{init}}$;

$\mathbf{x}_{\text{new}} := \infty$;

While[$|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}| > \varepsilon$]

{

$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} - \mathbf{g}(\mathbf{x}_{\text{old}}) / \mathbf{g}'(\mathbf{x}_{\text{old}})$;

}

return \mathbf{x}_{new} ;

(二)Code for BS Options

```
Function ImpVolCall(C As Double, K As Double, T As Double, S As Double, r As Double, f As Double) As Double
    Dim vol As Double
    Dim err As Double
    Dim dv As Double
    Dim d1 As Double, d2 As Double
    Dim price_err As Double
    Dim vega As Double

    vol = 0.2
    err = 0.0001
    dv = err + 1
    While ( Abs(dv)>err )
        d1 = (Log(S/K) + (r-f+vol*vol/2)*T)/(vol*Sqr(T))
        d2 = d1 - vol*Sqr(T)

        price_err = (S*Exp(-f*T)*NorCDF(d1) _
            - K*Exp(-r*T)*NorCDF(d2)) - C
        vega = S*Exp(-f*T)*Sqr(T)*Exp(-0.5*d1*d1)/Sqr(2*PI)
        dv = price_err/vega
        vol = vol - dv
    Wend

    ImpVolCall = vol
End Function
```

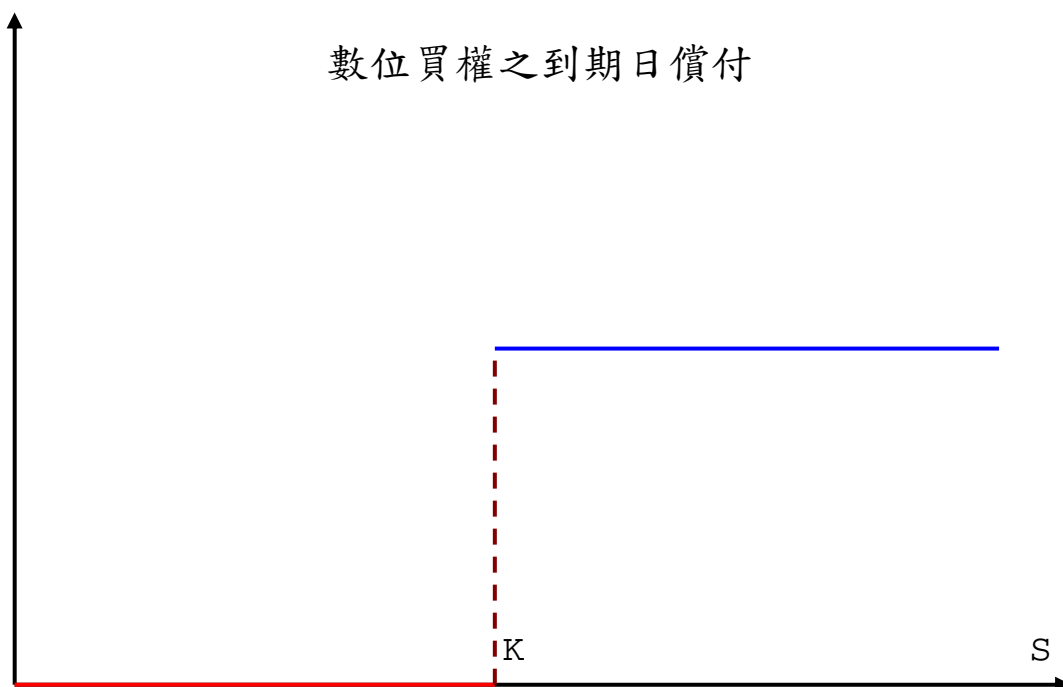

九、Exotic Option

(一) Digital Options

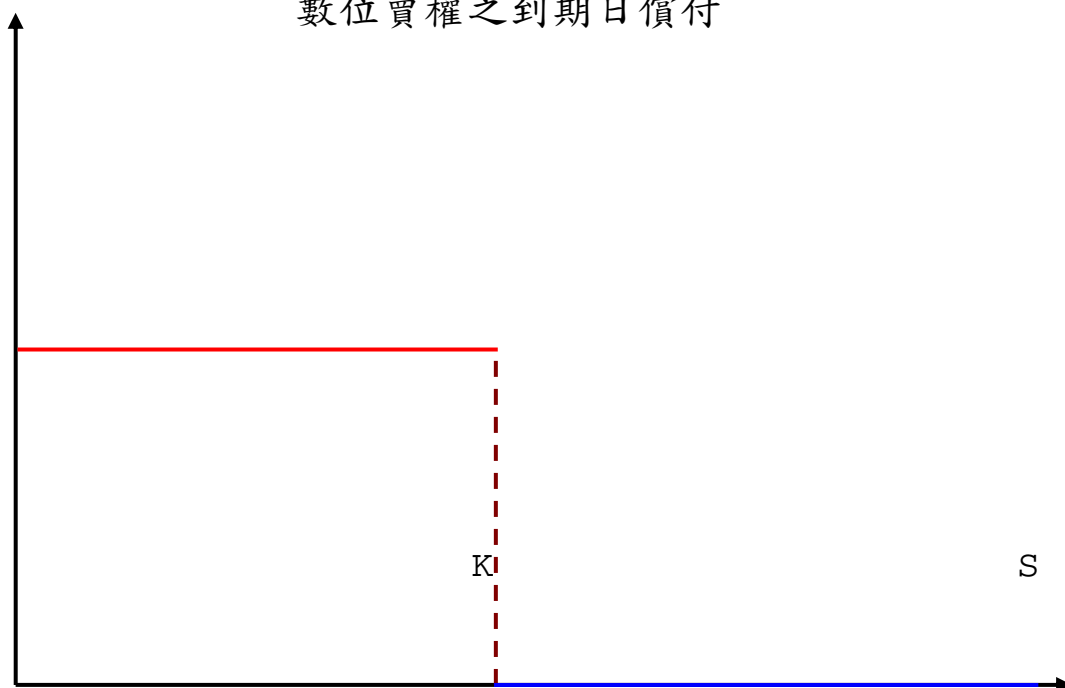
- ◆ Digital (Binary)：歐式數位選擇權在到期日時，如果處於實值情況則其償付為一元，否則沒有任何償付。
- ◆ 歐式數位買權由其到期日的償付定義之。今 T 表到期日時間， K 為執行價格，則其到期日償付可表示為，

$$DC_T = \begin{cases} 1, S_T > K \\ 0, S_T \leq K \end{cases}$$

- ☞ DC_T 表 T 時數位買權之權利金（償付額）。 S_T 為到期日標的資產價格。
- ☞ 償付圖形如下圖。注意，和一般選擇權不同，即使在極端實值之下，償付也是固定不變。



數位賣權之到期日償付



◆ 歐式數位賣權，其到期日的償付可表示如下

$$DP_T = \begin{cases} 1, S_T < K \\ 0, S_T \geq K \end{cases}$$

➤ 訂價公式

- ◆ 評定數位選擇權的方法與前述 BS 評價公式有相同的精神，即風險中立的評價方法。

☞ 我們可以利用二項式模型，或者類似 BS 模型的連續交易模型來求得訂價公式。

- ◆ 下面的公式即是在連續模型下導出的數位買權訂價公式

$$DC = e^{-rT} N(d - \sigma\sqrt{T})$$
$$d = \frac{\ln(S / K) + (r_f - r + \sigma^2 / 2)T}{\sigma\sqrt{T}}$$

☞ T：到期時間

☞ σ ：標的資產波動性

☞ r：無風險利率

☞ r_f ：外國無風險利率

- ◆ 亦可導出數位賣權公式

$$DP = e^{-rT} N(-d + \sigma\sqrt{T})$$

(二)Barrier Options

➤ Barrier Option：預先設訂上限和下限的阻隔

- ◆ 一旦資產價格越過(Knock)這些界限，阻隔選擇權便可能成為有效(In)或無效(Out)。
- ◆ 可能有 Rebate。

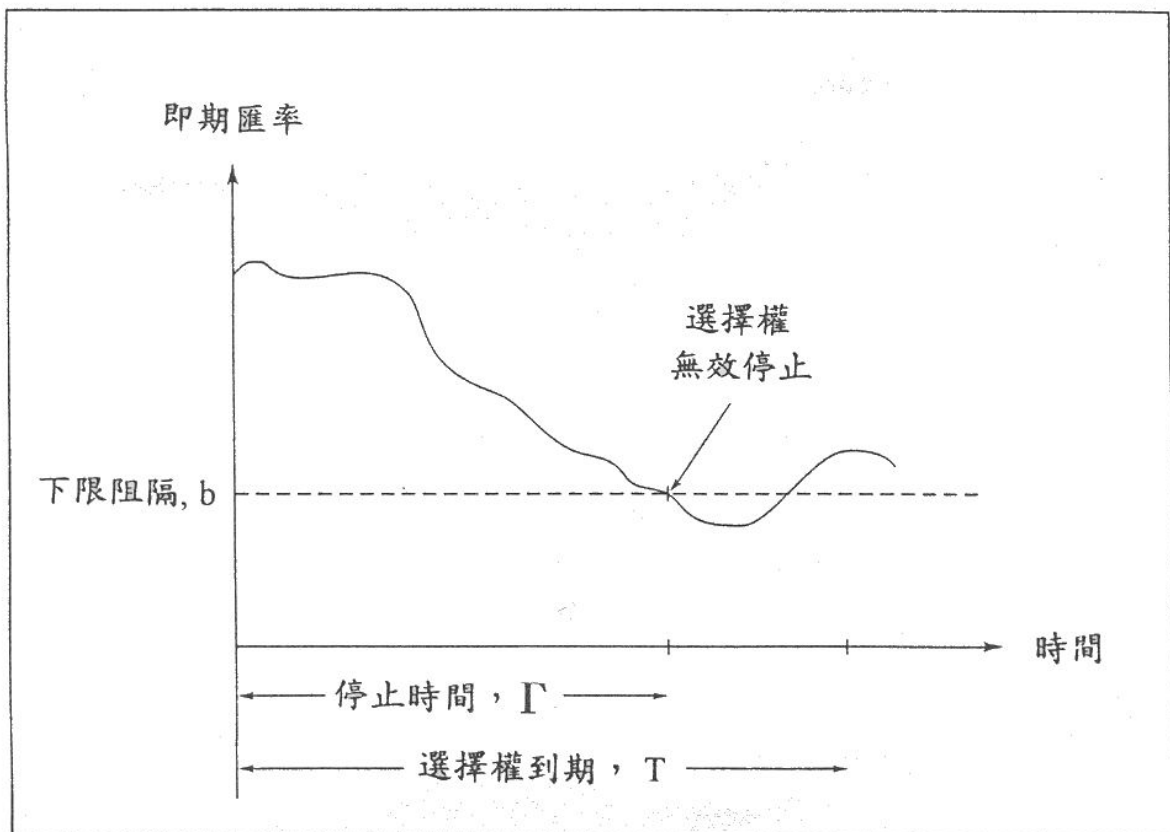
➤ 下限選擇權：分別為下限無效(down-and- out)、下限有效(down-and-in)

- ◆ 在下限無效選擇權中，下限阻隔已被確定。如果標的資產價格在選擇權的有效期間落於此下限阻隔之下，則此選擇權便無效停止。

☞ 在下圖中顯示了某一即期匯率(標的資產價格)的特定歷史。在這圖中，即期匯率兩度越過下限阻隔。當即期匯率第一次越過下限阻隔時，下限無效選擇權便停止失效了。

☞ 從目前到匯率第一次越過阻隔的時間，我們稱之為停止時間，以 Γ 表示。 Γ 本身是一個隨機變數，因為我們無法事前知道匯率越界的正確時間。引入停止時間這一概念，可以使我們在介紹阻隔選擇權上更為精緻。

即期匯率(標的資產價格)的歷史過程



◆ 令 T 表選擇權的到期日。定義下面的指標函數：

$I(\Gamma, T)$: 1, 阻隔沒有被穿越, $\Gamma > T$
: 0, 阻隔在 T 時間以前被穿越, $\Gamma < T$

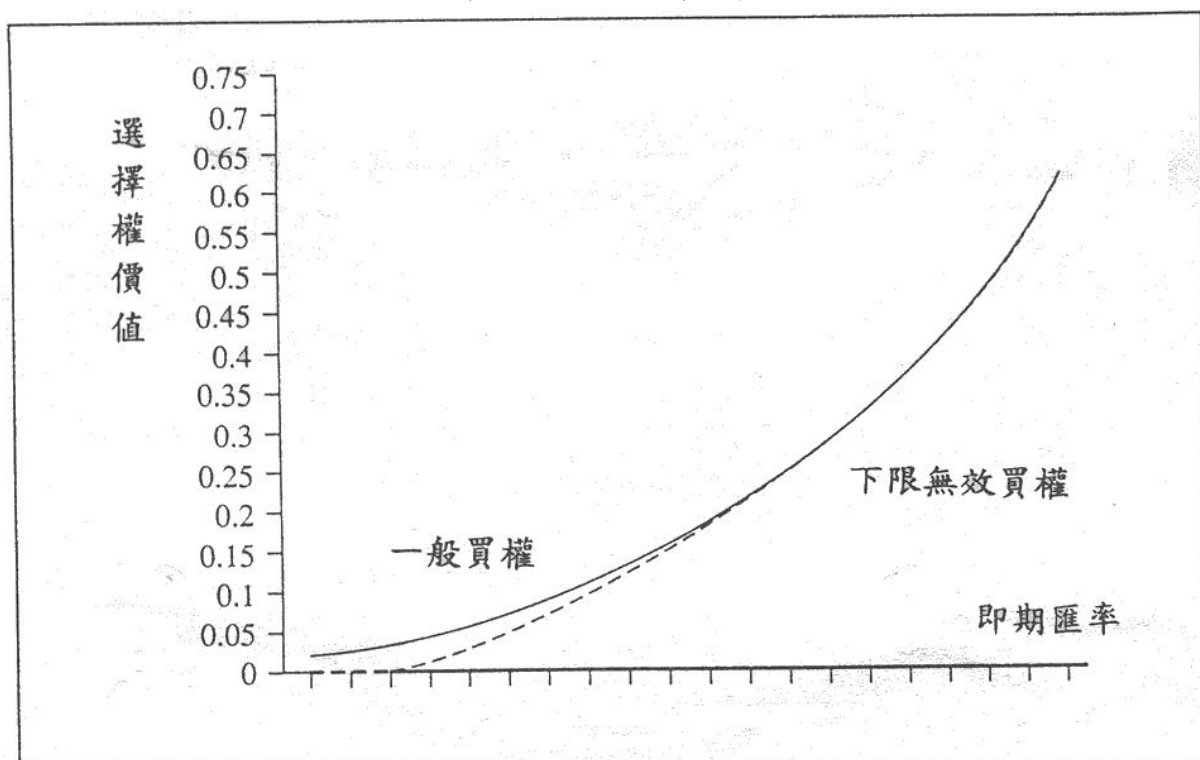
☞ 在上圖中 $\Gamma < T$ ，這是由於在到期前標的資產價格穿越了下限阻隔二次。

- ◆ 首先考慮下限無效買權。令 b 表下限阻隔，到期日為 T ，則執行價格為 K 之下限無效買權，在到期日的償付可表示為

$$DOC(S_T, 0, K, b) = \iota(\Gamma, T) \text{Max}\{S_T - K, 0\}$$

- 如果下限沒有被穿越， $\Gamma > T$ ，則 $\iota(\Gamma, T) = 1$ ，期末償付如同一般買權。如果下限被穿越， $\Gamma < T$ ，則 $\iota(\Gamma, T) = 0$ ，故償付為零，即選擇權消失了。
- 下限無效買入選擇權之價值不會大於相同 T 與 K 之一般買入選擇權。簡單地說，由於當標的資產價格低於 b 之下，這權利便消失。而在一般買權中，不論標的資產價格之高低為何，權利都存在。因此後者之權利範圍較大，故價值也會較高。

下限無效買權與一般買權價值之比較



◆ 下限無效買權之訂價公式，需依據 b, K 之關係分成兩部份討論：

❧ (i) $b \leq K$

$$\begin{aligned}
 SLBC(S_0, T) &= e^{-r_f T} S_0 N(d_1) - e^{-rT} K N(d_1 - \sigma \sqrt{T}) \\
 &\quad - \{e^{-r_f T} S_0 (b / S_0)^{2\eta+1} N(d_2) - e^{-rT} K (b / S_0)^{2\eta-1} N(d_2 - \sigma \sqrt{T})\} \\
 d_1 &= \frac{\ln(S_0 / K) + (r - r_f + \sigma^2 / 2)T}{\sigma \sqrt{T}} \\
 d_2 &= \frac{\ln(b^2 / (S_0 \bullet K)) + (r - r_f + \sigma^2 / 2)T}{\sigma \sqrt{T}} \\
 \eta &= \frac{r - r_f}{\sigma^2}, r \neq r_f
 \end{aligned}$$

❧ (ii) $b > K$

$$\begin{aligned}
 SLBC(S_0, T) &= e^{-r_f T} S_0 N(d_3) - e^{-rT} b N(d_3 - \sigma \sqrt{T}) + e^{-rT} (b - K) N(d_3 - \sigma \sqrt{T}) \\
 &\quad - \{e^{-r_f T} S_0 (b / S_0)^{2\eta+1} N(d_4) - e^{-rT} K (b / S_0)^{2\eta-1} N(d_4 - \sigma \sqrt{T})\} \\
 d_3 &= \frac{\ln(S_0 / b) + (r - r_f + \sigma^2 / 2)T}{\sigma \sqrt{T}} \\
 d_4 &= \frac{\ln(b / S_0) + (r - r_f + \sigma^2 / 2)T}{\sigma \sqrt{T}}
 \end{aligned}$$

- ◆ 考慮到期日 T ，執行價格 K 之下限無效賣權。到期日的償付可如下表示

$$DOP(S_T, 0, K, b) = \iota(\Gamma, T) \text{Max}\{K - S_T, 0\}$$

- ☞ 若下限阻隔沒有被穿越，則 $\Gamma > T$ ， $\iota(\Gamma, T) = 1$ ，因此和一般賣權無異；反之則 $\Gamma \leq T$ ， $\iota(\Gamma, T) = 0$ ，下限無效賣權便失效停止。至於其訂價公式，則和買權類似，在此不再列出。

- 上限選擇權：分別為上限有效(up-and-in)，以及上限無效(up-and-out)。
- Double Barriers：設定上限與下限兩個出、入局阻隔點。

(三) Paylater Options

➤ Pay Later：延後支付選擇權與一般選擇權的不同之處在於前者期初購入選擇權時並不立即支付權利金。只有當執行之後，延後支付選擇權的權利金才要支付。

◆ 此選擇權的特色在於延後支付權利金的權利。

☞ 在期末到期時，如果選擇權為實值的，則延後支付選擇權的買方，一定要執行選擇權，並且同時要付權利金，不論實值的程度是否大到足以負擔權利金。

☞ 如果到期時選擇權為虛值，則買方不但不會執行，同時不用支付權利金。延後支付選擇權一樣可分為兩種，即延後支付買權與延後支付賣權。

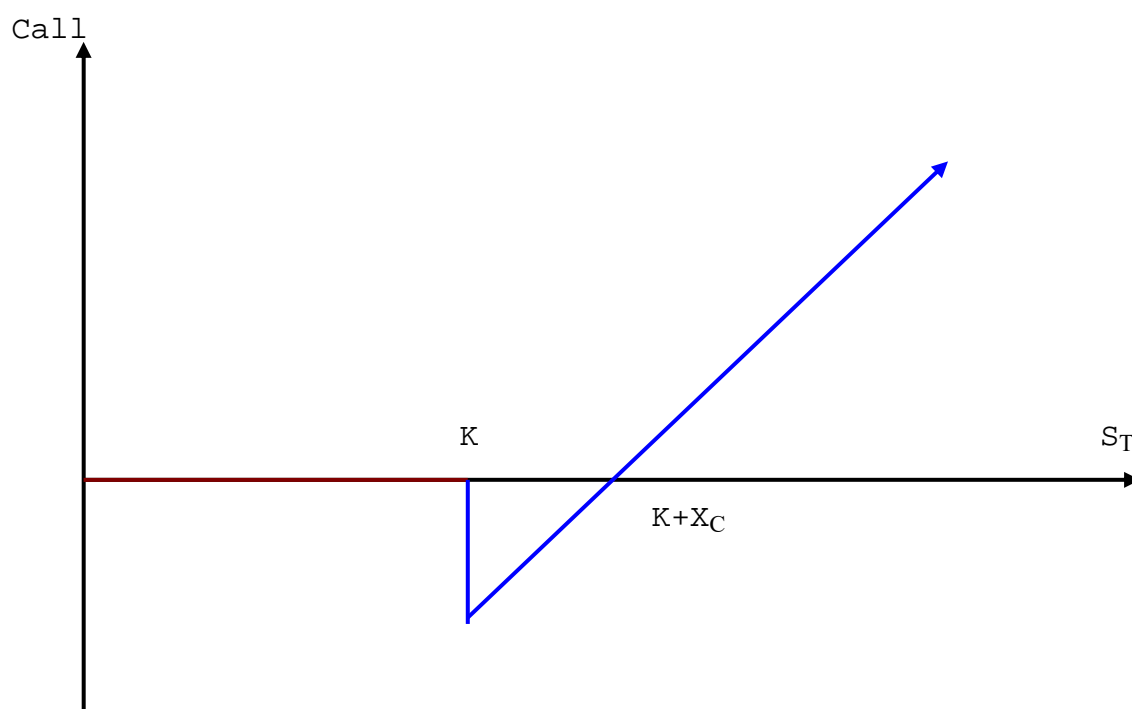
◆ 歐式延後支付買權由其到期日償付定義之。到期日之償付如下

$$PLC_T = \begin{cases} S_T - K - X_C, & S_T > K \\ 0, & S_T \leq K \end{cases}$$

☞ 其中 x_c 為此選擇權的權利金。

- ◆ 其償付如下圖所示。注意，如果到期日即期匯率介於 K 和 $K+X_c$ 之間，則償付為負的。此時標的資產價格大於 K ，但由執行選擇權所產生之現金流量卻不足以支付延遲的權利金。

延後支付買權到期日償付



- ◆ 一個延後支付選擇權可看成是由一個普通選擇權與 $-X_c$ 個數位選擇權的組合。我們可將前式分解，表示如下

$$PLC_T = \begin{cases} S_T - K \\ 0 \end{cases} - X_c \times \begin{cases} 1, (S_T > K) \\ 0, (S_T \leq K) \end{cases}$$

- ☞ 等式右邊第一項即為一般歐式選擇權之償付，第二項為 x_c 乘上一個歐式數位選擇權之償付。

- 所以一個延後支付選擇權，相當於一個歐式買入選擇權買入部位加上 x_c 個歐式數位買入選擇權之賣出部位的組合。
- 由於當延後支付選擇權在期初出售時買方是不需付錢的。
- 因此上式的期初價值為零。由此可得 x_c

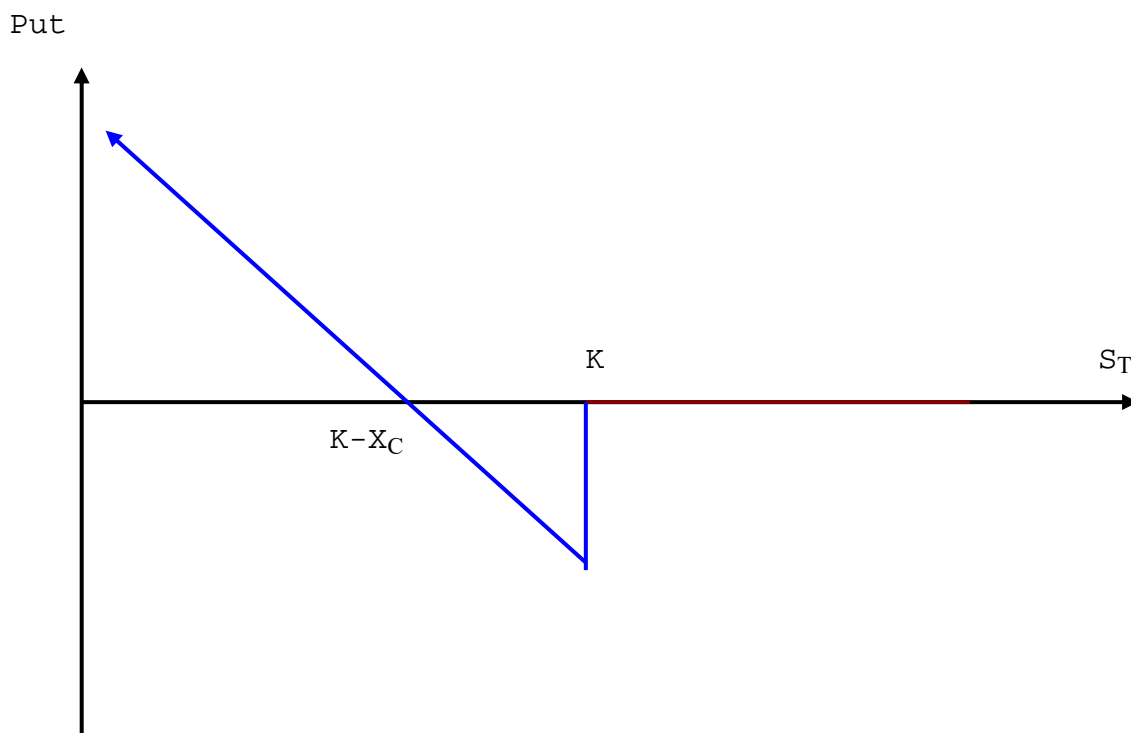
$$PLC = C - X_c \bullet DC = 0$$

$$X_c = \frac{C}{DC}$$

- 延後支付買權之價值為

$$PLC(S_t, T-t, K) = C(S_t, T-t, K) - X_c \bullet DC(S_t, T-t, K)$$

延後支付賣權到期日償付



◆ 歐式延後支付賣權到期日的償付定義如下

$$PLP_T = \begin{cases} K - S_T - X_p, & S_T < K \\ 0, & S_T \geq K \end{cases}$$

其中 x_p 為權利金。和前面相同的論點， t 時點延後支付賣權的價值為

$$PLP(S_t, T-t, K) = P(S_t, T-t, K) - X_p \bullet DP(S_t, T-t, K)$$

右式第一項為一般歐式外幣賣權價值，第二項則為歐式外幣數位選擇權之價值。

因此，一個延後支付賣權等於一個普通歐式賣權減去 x_p 個數位賣權。故此延後支付賣權之權利金為

$$X_p = \frac{P(S_0, T, K)}{DP(S_0, T, K)}$$

＋、VB Source Code for Option Pricing*

(一)Instrument Specifications

- Forward Start Options.....78
- Ratchet Options.....79
- Time-Switch Options.....79
- Chooser Options.....81
- Options On Options.....84
- Lookback Options.....86
- Barrier Options.....90
- Binary Options.....96
- Asian Options.....98

(二)VB Codes Function Prototypes

(三) Spreadsheets to Call VB Codes

- Implied Volatility Calculation
- Exotic Option Call Example
- Exotic Option Pricing Codes

敏感性可由差分比率求得

$$\text{delta} = \frac{\partial C}{\partial S} \approx \frac{C(S + \Delta S) - C(S - \Delta S)}{2\Delta S}$$

$$\text{gamma} = \frac{\partial^2 C}{\partial S^2} \approx \frac{C(S + \Delta S) - 2C(S) + C(S - \Delta S)}{\Delta S^2}$$

$$\text{vega} = \frac{\partial C}{\partial \sigma} \approx \frac{C(\sigma + \Delta \sigma) - C(\sigma - \Delta \sigma)}{2\Delta \sigma}$$

$$\text{theta} = \frac{\partial C}{\partial t} \approx \frac{C(t + \Delta t) - C(t - \Delta t)}{2\Delta t}$$

$$\text{rho} = \frac{\partial C}{\partial r} \approx \frac{C(r + \Delta r) - C(r - \Delta r)}{2\Delta r}$$

