

資產負債管理系統之開發

以 QuantLib 實作銀行簿利率風險管理

台中 靜宜大學 2019/5/16

昀騰金融科技

技術長

董夢雲 博士

dongmy@ms5.hinet.net

目 錄

- 一、前言
- 二、銀行資產負債表與風險管理全貌
- 三、傳統利率風險衡量的方式
- 四、現代利率風險衡量的方式
- 五、系統架構與特色
- 六、報表產出
- 七、無到期日存款模型介紹
- 八、提前還款模型介紹
- 九、NSFR & LCR
- 十、Basel III & IRRBB
- 十一、QuantLib 程式庫

昀騰金融科技股份有限公司

技術長
金融博士、證券分析師

董夢雲 Andy Dong



Line/WeChat: andydong3137
E: andydong1209@gmail.com
<https://github.com/andydong1209>
M: (T) 0988-065-751 (C) 1508-919-2872
10647 台北市大安區辛亥路一段 50 號 4 樓

學經歷

國立台灣大學電機工程學系學士

國立中央大學財務管理學研究所博士

中國信託商業銀行交易室研發科主管

凱基證券風險管理部主管兼亞洲區風險管理主管

中華開發金控、工業銀行風險管理處處長

永豐金控、商業銀行風險管理處處長

永豐商業銀行結構商品開發部副總經理

專業

證券暨投資分析人員合格(1996)

台灣金融研訓院 2019 年菁英講座

專長

風險管理理論與實務，資本配置與額度規劃、資產負債管理實務

外匯與利率結構商品評價實務，股權與債權及衍生商品評價實務

GPU 平行運算與結構商品系統開發，CUDA、OpenCL

CPU 平行運算與 ALM 系統開發，C#/C++/C、.Net Framework、SQL

人工智慧(Deep Learning)交易策略開發，Python、Keras、TensorFlow

一、前言

◆ 國際清算銀行的巴塞爾銀行監督管理委員會於 2016 年 4 月公布新版的銀行簿利率風險 (IRRBB)標準的新規範，其中，對於利率風險管理的要求有許多重大的改變。

- 銀行需同時以盈餘與經濟價值這兩個風險指標來表達風險胃納量。
- 這些指標需要在各種適當範圍的利率震盪及壓力情境下來衡量。
- 客戶行為選擇權對於銀行存、放款等產品契約現金流量的改變，必須加以謹慎考慮。

◆ IRRBB 標準的這些要求，與 IFRS 的改變在脈絡上是一致的。

- 以現值基礎來評估銀行的價值與風險，是一體兩面的事情。
- 金融資產價值評估的正確性，首要之處在於現金流量的掌握。
- 使用行為財務學與大數據的分析，來合理評具客戶行為權利的存、放款的現金流量實質落點，是正確計算資產價值與風險的基礎。

◆ 如何落實 IRRBB 的要求，對於銀行而言，有其高度的挑戰性。

- 分享講師之前在銀行界內部自行開發 ALM 系統的經驗。
- 該系統可以計算銀行簿的盈餘(NII)與經濟價值(EV)，以及模擬利率變動下的 ΔNII 與 ΔEVE 。
- 該系統內建多因子遠期利率隨機過程(Libor Market Model)的模擬產生器，可以進行銀行簿利率風險模擬的分析計算。
- 該系統建置了分析客戶行為選擇權的量化模型，包括不動產放款提前還款模型以及活期存款提領模型。

二、銀行資產負債表與風險管理全貌

(一)銀行的資產負債表

資金用途				資金來源			
項目	利率	均量	比重	項目	利率	均量	比重
Cash/存放同業 存放央行 拆放同業 應收/付遠匯款				同業拆放/RP 應收/付遠匯款			
買入有價證券 央行 NCD 票券 債券 放款 短期 中長期 Factoring				存款 活期 定期			
長投 固定資產 其他				金融債 其他 資本&公積			

◆ 如何衡量與管理整體部位的風險？

➤ Basel 的風險管理架構觀點

- ✓ 市場風險
- ✓ 信用風險
- ✓ 作業風險
- ✓ 流動性風險

(二)銀行的簿別分類

◆ 交易簿：為交易目的持有之部位，是指意圖短期持有以供出售，或是意圖從實際或預期之短期價格波動中獲利或鎖定套利利潤。

➤ 如自營部位、代客買賣(如撮合成交之經紀業務)與創造市場交易所產生之部位。

➤ 就持有目的而言，可列屬交易簿之部位可歸納如下：

- ✓ 1.意圖從實際或預期買賣價差中賺取利潤所持有之部位。
- ✓ 2.意圖於從其他價格或利率變動中獲取利潤所持有之部位。
- ✓ 3.因從事經紀、自營業務所持有之部位。
- ✓ 4.為抵銷交易簿上另一資產部位或投資組合之全部或大部分風險，而持有之部位。
- ✓ 5.所有可逕自於預定投資額度內從事交易之部位。

➤ 以上部位並須在交易方面不受任何契約條款限制，或者可完全進行風險規避。

◆ 銀行簿：不屬交易簿之部位者，列為銀行簿之部位。

◆ 34 號公報、IFRS 與市價重估的趨勢

- 針對金融資產 34 號公報採用五類分類，公平價值(Fair Value)、備供出售(Available for Sale)、持有至到期(Hold to Maturity)、攤銷後成本衡量、成本衡量。
- IFRS 採用兩類分類，公允價值衡量、攤銷後成本衡量。
 - ✓ 固定收益資產與金融債使用公允價值衡量。
 - ✓ 放款與存款傾向使用攤銷後成本衡量。
- 市場潮流為盡可能使用公允價值衡量來評估所有資產與負債項目。

(三)銀行的風險管理架構

◆ 交易簿的管理

- 以市場風險涉險值(Market Risk Value at Risk, MVaR)的方法計算交易部位的風險。

◆ 銀行簿的管理

- 以信用風險涉險值(Credit Risk Value at Risk, CVaR)的方法計算放款與投資部位的風險。

◆ 存放利率與流動性的管理

- 以 ALM 系統來控管流動性與利率缺口(Interest Rate Gape)的風險。
- 利率缺口起源於存、放款不同期限所致。

三、傳統利率風險衡量的方式

(一)流量的觀點—利息收入的影響

◆ 利率變動對存款與放款的利息皆有影響。

- 定義利率敏感性資產(RSA)為一定天期內(一年)會重設其利率之資產。
- 定義利率敏感性負債(RSL)為一定天期內(一年)會重設其利率之負債。
- 定義重定價缺口(GAP)為如下，

$$GAP = RSA - RSL$$

◆ 從會計盈餘的觀點，計算利率變動對利息收入的影響。

- 淨利息收入(NII)因利率變動的影響可表示如下，

$$\Delta NII = GAP \times \Delta R$$

- 若一年內的 $GAP = 1$ 億，則 0.25% 的利率變動，對 NII 的影響為 25 萬。

$$100,000,000 \times 0.0025 = 250,000$$

◆ 只考慮一年內利息盈收的效果，一年以上沒有考慮。

- 價值的變動沒有反應出來。

(二)存量的觀點—價格的影響

◆ 利率的變動對所有固定收益證券的價值都有影響。

- 以放款為例，放款利息的影響不是只有一年內的效果。
- NII 的計算只考慮了一年內的現金流量效果。
- 一年以上的 Cash Flow 與利率改變的 Discount Effect 並沒有考慮到。
- 會計 34 號公報的修正，要求對銀行簿市價重估。

◆ 若要正確評估其效果，需由價值的變動著手。

- 公司價值可表示如下，

$$E = A - L$$

- 將資產與負債視為固定收益證券，利用 Duration 公式如下，

$$\frac{\Delta P}{P} = -MD \times \Delta R$$

- 一 MD = 10 年的債券，目前價格為 1,000，利率變動 25b.p.下，價格變動量為，

$$\Delta P = -1,000 \times 10 \times 0.0025 = 25$$

◆ 公司因利率變動而產生的價值變動，可表示如下，

$$\Delta E = \Delta A - \Delta L = (-A \times MD_A \times \Delta R) - (-L \times MD_L \times \Delta R)$$

$$= -E \times \left(\frac{A}{E} \times MD_A - \frac{L}{E} \times MD_L \right) \times \Delta R$$

$$= -E \times MD_E \times \Delta R, \quad MD_E = \frac{A}{E} \times MD_A - \frac{L}{E} \times MD_L$$

- 一但求得資產 Duration， MD_A ，與負債 Duration， MD_L ，便可求得權益之 Duration Gap， MD_E 。
- 我們便可求得公司價值因利率變動而產生的影響。

◆ 利率風險中立下的情況，

- 當 $MD_E = 0$ 時，公司為利率風險中立。
- 此時的所採行的交易策略為免疫(Immunization)策略。

◆ Duration 公式假設利率期限結構為平行移動，此與市場實際情況不符。

- 資產、負債表上的改變並不容易，Immunization 的執行並不容易。
- 資產、負債中有許多非債券型的利率衍生商品，無法定義 Duration。
- 實務上使用有限制，較無法成為避險操作的根據。

四、現代利率風險衡量的方式

(一)流量的觀點—時間帶的價格變動

◆ 利率敏感性工具，其現金流量為利率之函數。

- 此現金流量是有時間屬性的，可能為 0~1M，1M~3M，3M~6M，6M~12M。
- 現金流量的價值需以當時之利率折現之。

◆ 以一個面值 100 元之 5 個月零息債為例，現金流量落於 3M~6M 之時間帶， $CF_0 = 100$ 。

- 以目前之利率將 CF_0 折現，求得其目前價值得 DV_0 。
- 若利率上漲 1b.p.，其現金流量成為 CF_1 。此例中 $CF_1 = CF_0$ ，但並不必然如此，例如 FRA。
- 以改變後之利率將 CF_1 折現，求得其目前價值得 DV_1 。

◆ 定義 $DV01 = DV_1 - DV_0$ ，表 1b.p.利率變動下，金融工具價格的變動量。

➤ 若 1M~3M 時間帶中的 DV01 為 100 萬，則若此時間帶中利率下跌 25b.p.，損益變動約為，

$$PL = DV01 \times \Delta R = 1,000,000 \times 0.0025 = 2,500$$

➤ 允許利率期限結構非平行移動，分段考量損益變化，再加總合併估計總損益。

◆ 一個 DV01 缺口的示範。

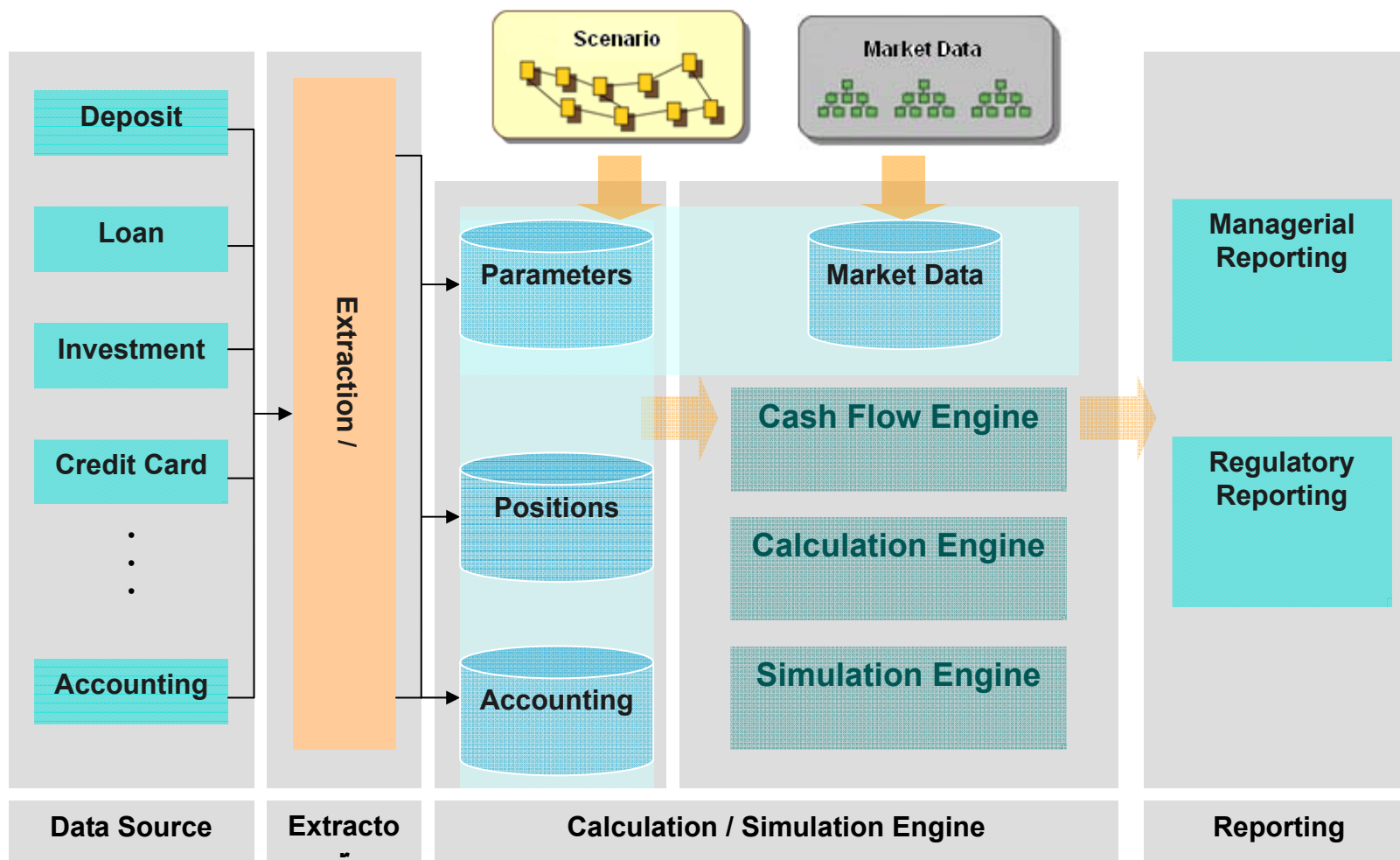
	0~1M	1M~3M	3M~6M	6M~12M	12M~24M
DV01	-2,000	1,000	-3,000	5,000	10,000
$\Delta R(\text{bp})$	-50	-40	-25	-25	-12.5
PL	100,000	-40,000	75,000	-125,000	-125,000
Total PL	-115,000				

◆ 對於任一時間帶中的利率風險，可單獨分離出來。

- 任何利率敏感工具，皆可計算 DV01。
- DV01 為線性的風險衡量。
- 利於避險操作。

五、系統架構與特色

(一)系統架構



◆ 開發平台

- Windows .Net Platform , Visual C#
- QuantLib Library in C#
- MS SQL Server

◆ Market Data 畫面

[圖此處]

◆ Position 畫面

[圖此處]

(二)系統特色

- ◆ 提高報表產生自動化程度
- ◆ 提供管理階層盈餘觀點及經濟價值(Economic Value)觀點利率風險分析資訊
- ◆ 考慮無到期日存款之行為(Behaviorization)分析
- ◆ 考慮放款之提前還款(Prepayment)行為分析
- ◆ 提供模擬分析工具(Monte Carlo Simulation)

◆ 採用更細緻之衡量風險方法

- 時間帶可再細分，不受法報格式限制
- 產品項目可展開至細項(ex：放款→企金、個金→房貸、車貸、信貸、其他)
- 提高現金流量之完整性，不僅考慮本金，也考慮利息
- 除 Risk-free Yield Curve 外，並依據金控信用評等，建構 9 條不同利率加碼之 Yield Curve，可針對存放款作市價重估
 - ✓ 首先利用 Risk-free Yield Curve 求算出 Risk-free Spot Rate Curve
 - ✓ 利用台灣經濟新報(TEJ)與 S&P 所提供各評等的違約機率，轉換為信用風險貼水
 - ✓ 金控內部評等與 TEJ 和 S&P 做一對應關係，在計算各產品的信用風險貼水時，台幣可參考 TEJ；外幣可參考 S&P
 - ✓ 在求算各信用等級的即期利率期限結構，可加入各產品信用風險貼水，以利評價

六、報表產出

(一)法定報表

◆ 利率風險

- AI.230 利率敏感性資產負債分析表---新台幣
- AI.235 利率敏感性資產負債分析表---美金(OBU+DBU)

◆ 流動性風險

- AI.240 新台幣到期日期限結構分析表
- AI.245 美金到期日期限結構分析表(OBU+DBU)

◆ 利率風險

- AI.230 利率敏感性資產負債分析表---新台幣

[表此處]

➤ AI.235 利率敏感性資產負債分析表---美金(OBU+DBU)

[表此處]

◆ 流動性風險

- AI.240 新台幣到期日期限結構分析表

[表此處]

➤ AI.245 美金到期日期限結構分析表(OBU+DBU)

[表此處]

(二)管理性報表

◆ NII 相關

- Repricing Gap
- NII Impact

◆ Liquidity 相關

- Liquidity Report

◆ Economic Value 相關

- Cash Flow Maturity Gap
- DV01
- Economic Value(Present Value)
- Duration

◆ NII 相關

➤ Repricing Gap :

$$GAP = RSA - RSL$$

- ✓ 以會計帳面價值和銀行簿為基礎，並假設資產、負債均持有至到期日，RSA 與 RSL 的缺口或比率
- ✓ 利率敏感性資產(RSA)：在一定天期內，因利率變動會被重新訂價的資產總和
- ✓ 利率敏感性負債(RSL)：在一定天期內，因利率變動會被重新訂價的負債總和
- ✓ 重訂價缺口(GAP)：利率敏感性資產(RSA)與利率敏感性負債(RSL)之差額

➤ Repricing Gap Report

[表此處]

- NII Impact：從會計盈餘的觀點，計算利率變動對淨利息收益的影響

$$\Delta NII = GAP \times \Delta R = (RSA - RSL) \times \Delta R$$

- ✓ 舉例說明：假設一年內的重訂價缺口為+1,000 萬元，當利率上升 0.25%，淨利息收益將增加約 25,000 元

$$\Delta NII = GAP \times \Delta R = 10,000,000 \times 0.0025 = 25,000$$

➤ NII Impact Report

[表此處]

◆ Liquidity 相關

- Liquidity Gap：以會計資產負債表之帳面價值為基礎
 - ✓ 僅包含表內項目
 - ✓ 自編表基準日至會計科目細目預計到期日，來決定擺放天期
 - ✓ 與 AI.240 相同，只包含名目本金與下一次應計利息

➤ Liquidity Report

[表此處]

◆ Economic Value 相關

➤ Cash Flow Maturity Gap：依據財務理論及不同契約條件推算未來現金流量

- ✓ 僅考慮銀行簿與利率敏感相關的資產與負債
- ✓ 不僅考慮名目本金與下次應計利息，更考慮到各契約到期日前所有現金流量
- ✓ 放款方面，同時考慮到不同本金攤還方式與每期可能的提前還款情形
- ✓ 在活期存款方面，同時設算未來每期可能的存款流失情形

➤ Cash Flow Maturity Gap Report

[表此處]

➤ DV01：當利率變動 1bp 下，金融工具價格的變動量

$$DV01 = V' - V$$

V' ：利率上升 1bp，金融工具的價格

V ：利率不變下，金融工具的價格

✓ 舉例說明：假設某金融工具目前的 MTM 為 100 萬，當利率上升 1bp 下，價格變為 99.9 萬，則金融工具的 DV01 為 -0.1 萬

$$DV01 = V' - V = 99.9\text{萬} - 100\text{萬} = -0.1\text{萬}$$

➤ DV01 Report

[表此處]

- Economic Value Report：給定不同利率情境下，資產與負債經濟價值的變化

[表此處]

➤ Duration：資產與負債價值受利率影響的敏感程度

✓ 淨值存續期間(D_E)計算公式：

$$D_E = \frac{D_A \times A - D_L \times L}{A - L}$$

✓ 公式推算：

$$\frac{\Delta E}{E} = - \left[\frac{D_A \times A - D_L \times L}{A - L} \right] \times \frac{\Delta R}{1 + R} = -D_E \times \frac{\Delta R}{1 + R}$$

✓ 求算出 D_E 後，可算出當利率變動時，對淨值的影響程度

➤ Portfolio Duration Report

[表此處]

✓ 以銀行 D_E 為-14.46，當利率上漲 25bps，權益價值增加百分比為 3.62%

$$\frac{\Delta E}{E} = -D_E \times \Delta R = -(-14.46) \times 0.0025 = 3.62\%$$

七、無到期日存款模型介紹

◆ 估計存戶每期提領後，帳上所剩餘的金額

◆ 使用模型：OAS Model (Option-Adjusted Spread Model)

$$W_t = \beta_0 + \beta_1 Avg3mCP_{over6M} + \beta_2 5yIRS_{t-2} + \beta_3 I(quarter_t) + \varepsilon_t$$

✓ W_t ：第 t 期存款餘額

✓ $Avg3mCP_{over6M}$ ：過去 6 個月 3M CP rate 平均值

✓ $5yIRS_{t-2}$ ：前 2 個月 5Y Swap rate

✓ $I(quarter_t)$ ：指標函數， $I(quarter_t) = \begin{cases} 1 & quarter_t = Q1 \\ 0 & others \end{cases}$

➤ 針對 $\beta_1 \sim \beta_3$ 進行估計(線性迴歸估計)

✓ 估計結果如下： $\beta_1 < 0, \beta_2 > 0, \beta_3 > 0$

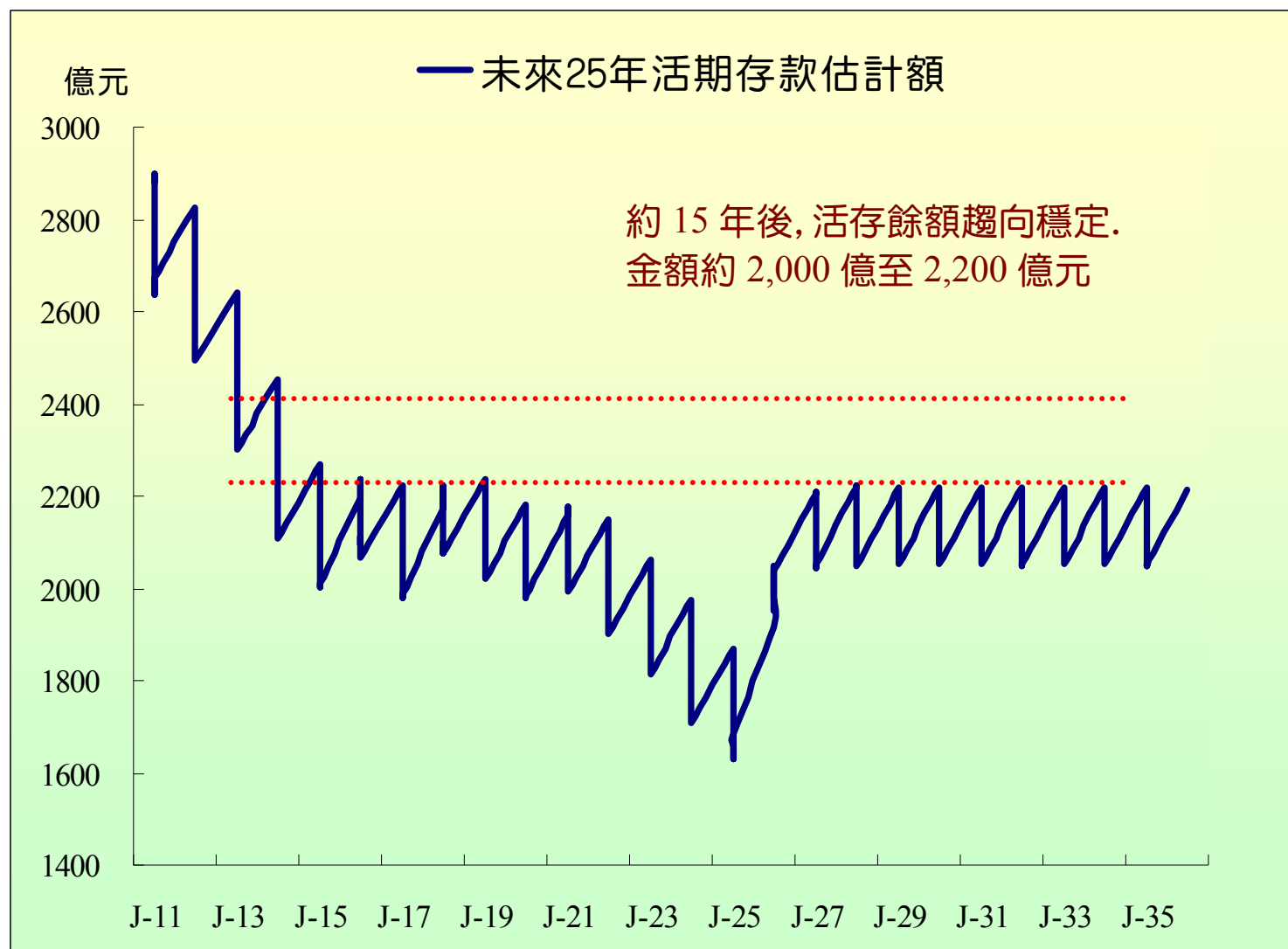
◆ 實際計算運用

- 用於非定存類的存款(含支存、一般存款、儲蓄存款)
 - ✓ 利用歷史資料估算參數模型
 - ✓ 直接利用模型估計未來各期之活期存款餘額，與上期的差額即是本期流出的存款金額

◆ 參考書目

- Martin M. Bardenhewer， “Modeling Non-maturing Products” (2006)。

◆ 參數估計結果



八、提前還款模型介紹

- ◆ 使用 OTS(Office of Thrift Supervision) Model，利用各期實際餘額及依原始契約條件繳款的餘額，計算提前還本率， CPR_t 。

$$CPR_t = seasoning_t \times seasonality_t \times refi_t$$

- $seasoning_t$ ：趨勢因子

$$seasoning_t = \begin{cases} 1/30 + (29/30) \times t/30 & , t \leq 30 \\ 1 & , t > 30 \end{cases}$$

- ✓ t ：該筆放款目前期數

- $seasonality_t$ ：季節性因子

$$seasonality_t = 1 + \beta_1 \times \sin \left\{ \beta_2 \times \left[\frac{month + t - 3}{3} \right] - 1 \right\}$$

- ✓ $month$ ：發生月份(此處指初貸日期)

➤ $refi_t$ ：再融資刺激因子

$$refi_t = \beta_3 + \beta_4 \times \arctan \left\{ \beta_5 \times \left[\beta_6 - \frac{c}{m_{t-3}} \right] \right\}$$

✓ m_{t-3} ：外部利率(前3期)

✓ c ：該筆放款目前適用利率

➤ 針對 $\beta_1 \sim \beta_6$ 進行估計(非線性最適化)

◆ 實際計算應用

➤ 利用歷史資料估算模型參數

➤ 每期計算現金流量時，帶入變數資料如下：

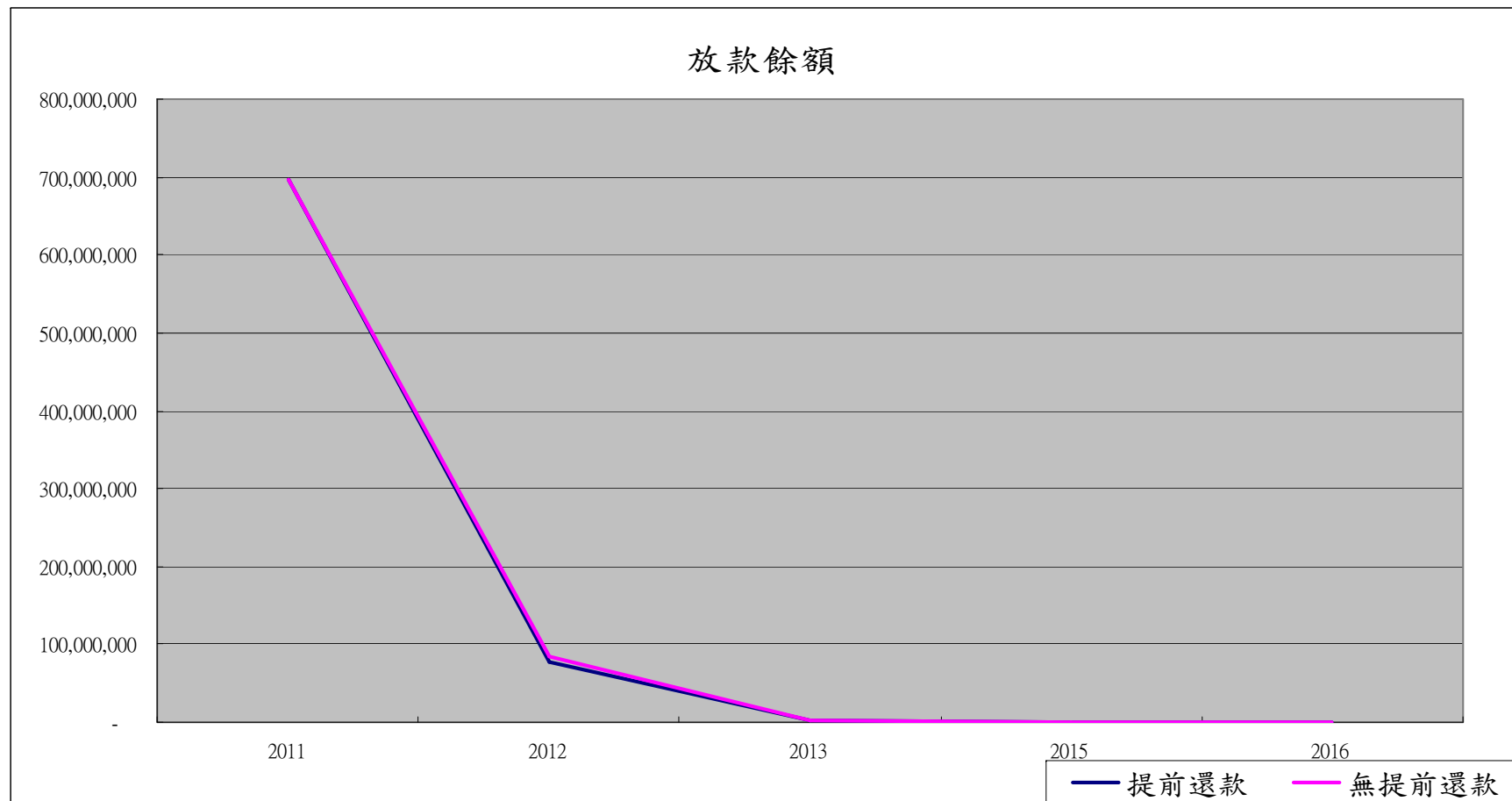
✓ 初貸日期、貸款期數、客戶目前貸款利率及當下外部房(信、車)貸利率以計算提前還款率

➤ 當期本金在扣除正常還款金額及提前還款金額，剩餘本金再納入下期利息計算

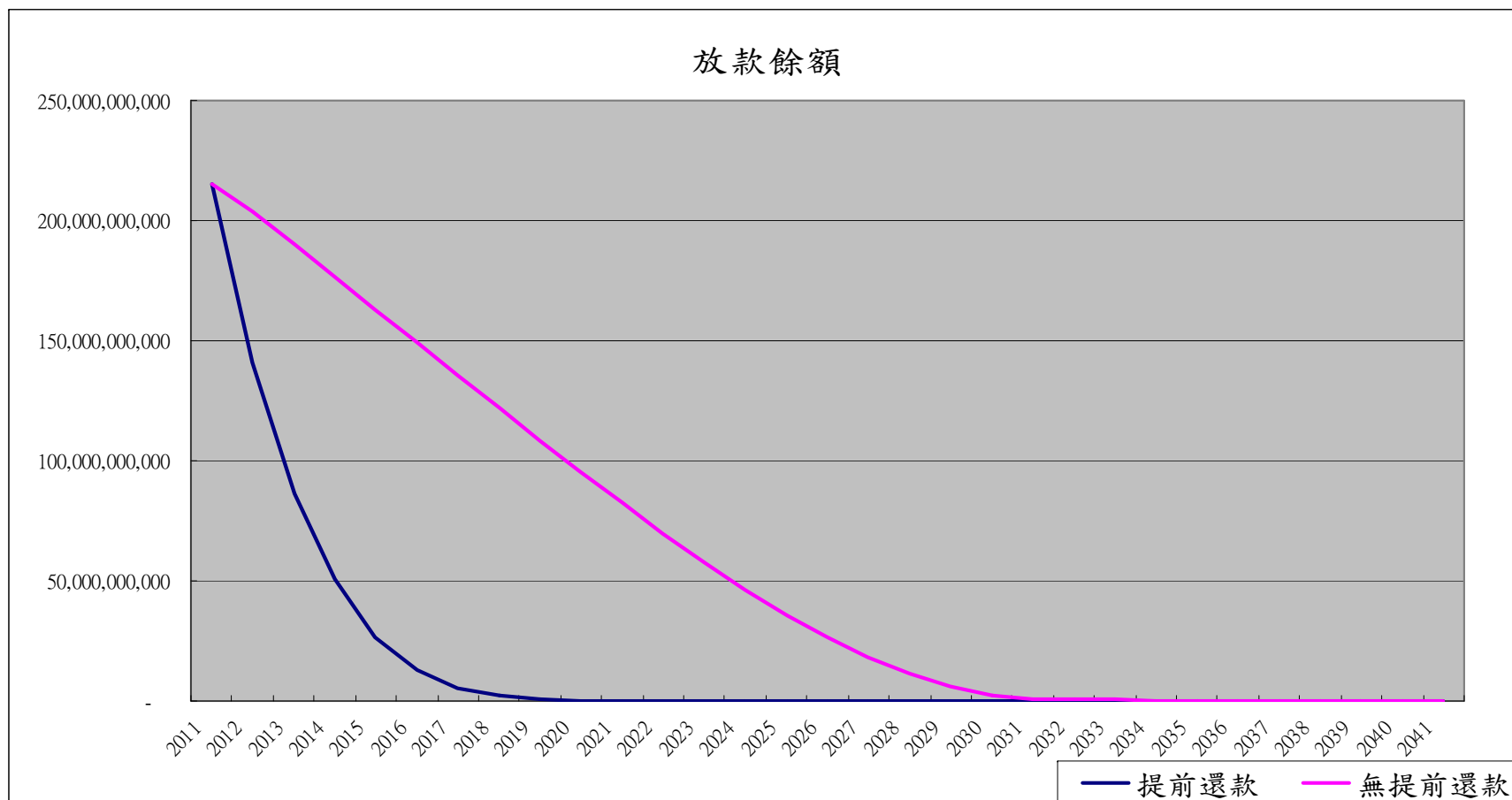
➤ 最後將各期現金流量折現求算放款現值

◆ 參數估計結果

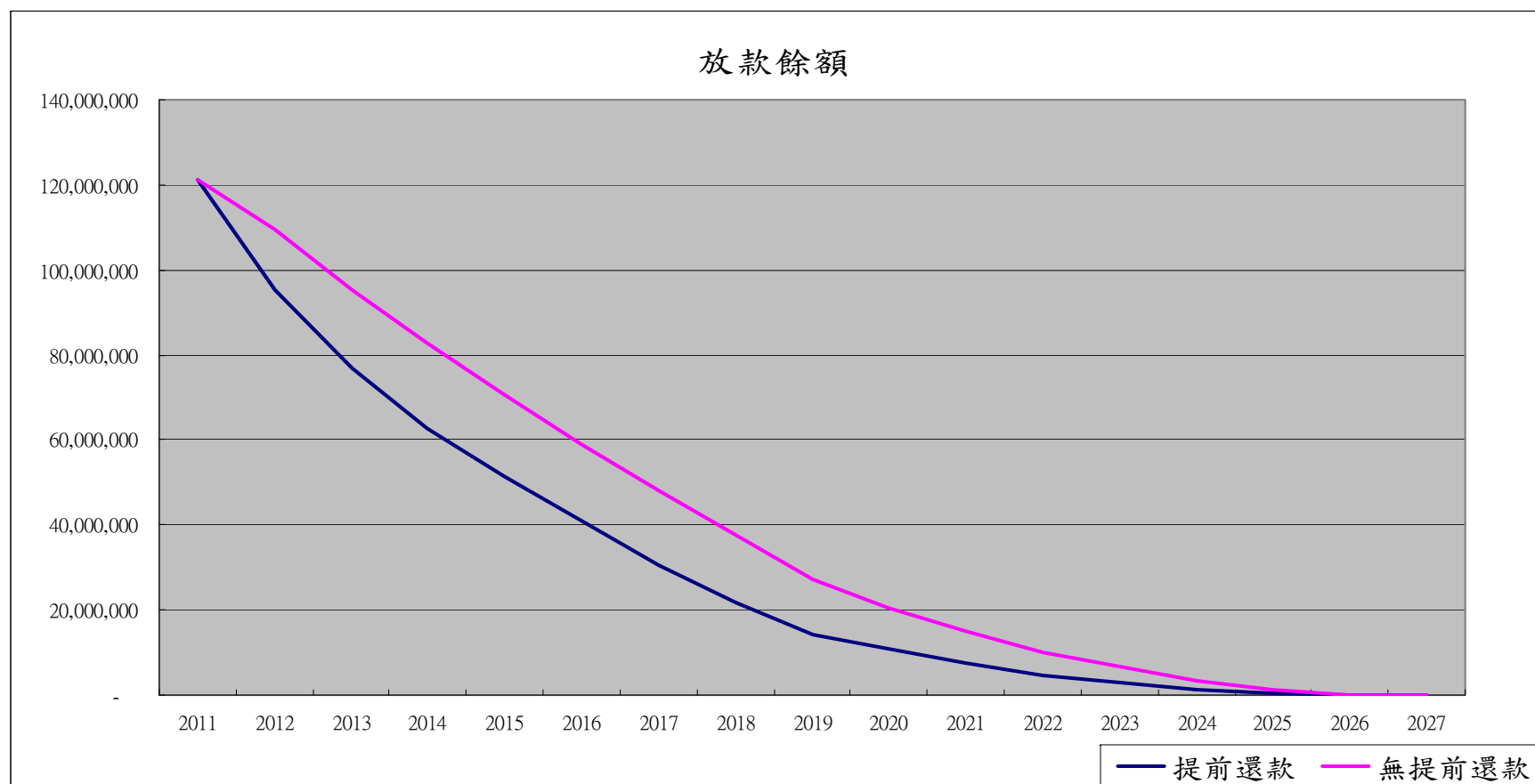
- 不同還本方式下，提前還款有無之圖形比較
- 房貸-到期還本



➤ 房貸-本息均攤



➤ 房貸-本金均攤



九、LCR & NSFR

(一) LCR(Liquidity Coverage Ratio)

◆ 為強化銀行短期流動性復原能力，銀行應確保持有足夠未受限制之合格高品質流動性資產，以因應壓力情境下 30 天內之淨現金流出。

◆ 計算方法：

$$\text{流動性覆蓋比率} = \frac{\text{合格高品質流動性資產總額}}{\text{未來 30 個日曆日內之淨現金流出總額}} \times 100\%$$

◆ 流動性覆蓋率（LCR）建立在傳統的流動性“覆蓋率”方法基礎之上，該方法在銀行內部被用來評估偶發流動性事件可能導致的風險暴露。

- 該壓力情景下的資金累計淨流出量應該按照未來 30 天計算。流動性覆蓋率的標準是不低於 100%（即高流動性資產至少應該等於估算的資金淨流出量）。
- 各家銀行應該持續滿足這一要求，並持有無變現障礙和高流動性資產儲備，用來抵禦可能發生的嚴重流動性壓力。
- 銀行和監管機構也要注意任何潛在的 30 天期限錯配情況，並確保有足夠的流動性資產來填補這一個月當中的資金缺口。

(二) NSFR(Net Stable Fund Ratio)

◆ 為促進銀行業機構的資產和業務融資更趨中、長期化。

- 根據銀行在一個年度內資產和業務的流動性特徵設定可接受的最低穩定資金量，作為一個強制執行的最低要求。
- 流動性覆蓋率指標的一個補充，鼓勵銀行通過結構調整減少短期融資的期限錯配、增加長期穩定資金來源，提高監管措施的有效性。

◆ 計算方法：

$$\text{淨穩定資金比率} = \frac{\text{可用的穩定資金}}{\text{業務所需的穩定資金}} \times 100\%$$

◆ NSFR 方法是建立在傳統的“淨流動性資產”和“資本金”計算方法之上的，這些傳統方法為國際銀行機構、銀行分析師和評級機構所廣泛使用。

- NSFR 並不局限於傳統方法，將傳統公式所忽略的表外資產潛在的流動性風險，以及用短期資金為長期資產融資等期限錯配的情況，都納入了考慮範圍。
 - ✓ 一種流動性風險的綜合計量方法，可以識別當前市場面臨的困難，包括在流動性不足的市場環境下正在進行之中的交易及證券化行為導致的融資需求。
- 在計算有穩定融資需求的資產總量時，應將所有缺乏流動性的資產和證券囊括在內，
 - ✓ 不管其在會計處理上的分類如何（無論其為交易類資產，還是可供出售資產或持有到期資產），也不管其交易及證券化行為何時完成。
- 實際上，有多少交易資產需要穩定資金來源，不是根據設定的執行週期，而是根據所持資產頭寸的流動性特徵來計算的。
 - ✓ 至少還有一部分穩定資金來源，要被用於滿足部分表外業務或突發事件潛在的流動性需求。

十、Basel III & IRRBB

◆ 銀行簿利率風險（以下簡稱 IRRBB）是巴塞爾資本架構第二支柱（監理審查程序）的一部分，需遵循委員會於 2004 年發佈之「利率風險管理與監理原則」（以下簡稱「利率風險原則」）指引。

➤ 此利率風險原則列出委員會對銀行在辨識、衡量、監督和控制 IRRBB，以及相關監理的期望。

◆ 依 FSB 的 TLAC 條款清單第 15 節所述：

➤ 為因應市場與監理實務的改變，委員會決定更新初版利率風險原則，本文件涵蓋委員會期許銀行用以衡量、管理、監督和控制該風險之原則與方法的更新版本。

➤ 這些更新的原則是 2015 年諮詢的主題，當時委員會提出了 IRRBB 監管處理的兩個選項：

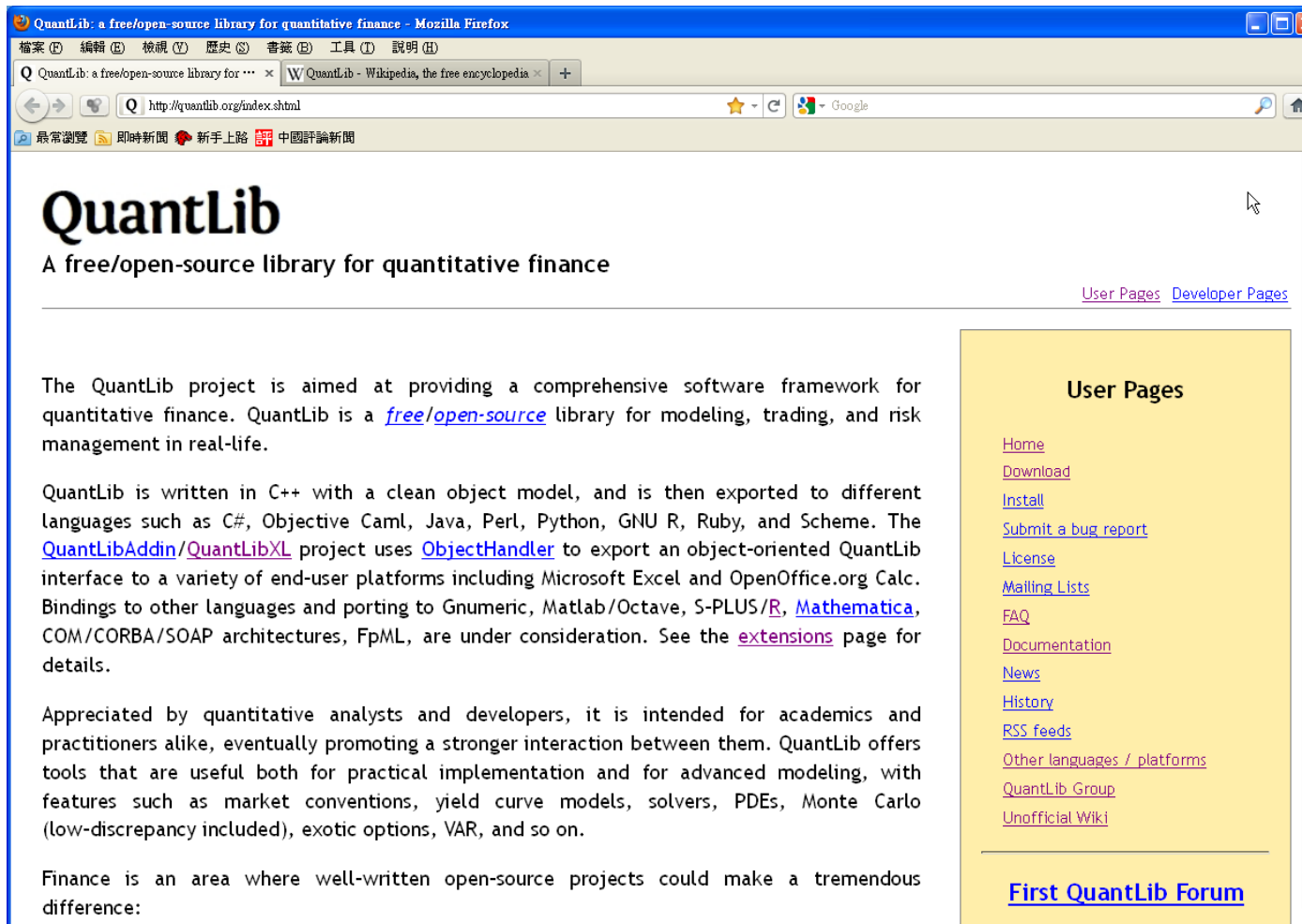
- ✓ 標準化的第一支柱（最低資本要求）方法論和強化的第二支柱方法論（包括第三支柱的要素-市場紀律），
- ✓ 委員會注意到業界對第一支柱標準法對 IRRBB 可行性的回饋意見，特別是制定 IRRBB 標準化措施所涉及的複雜性，該標準化衡量方式需準確且具風險敏感性，以反映制定法定資本要求的意涵。

◆ 在強化第二支柱下，對原則的主要更新如下：

- 對銀行 IRRBB 管理程序的期望提供了更佳的指引，尤其是發展衡量 IRRBB 時所運用的震盪和壓力情境（原則 4）、衡量 IRRBB 時銀行應該考量的關鍵行為和模型假設（原則 5），以及銀行 IRRBB 所使用的內部衡量系統（IMS）及模型之內部驗證程序（原則 6）。
- 原則 8 的揭露要求已經更新，以提高 IRRBB 衡量和管理的一致性、透明度及可比性。
- 除其他要求外，銀行必須揭露利率震盪對其權益經濟價值（ ΔEVE ）和淨利息收入（ ΔNII ）變化的影響，這是根據一系列規定的利率震盪情境計算出來的。
- 原則 11 更新監理審查程序，以闡明監理機關在評估銀行 IRRBB 暴險程度和管理應考量的因素。監理機關亦可要求司法管轄區內的銀行遵循 IRRBB 的標準化架構（例如，若發現銀行的 IMS 沒有充分反映 IRRBB）。標準化架構已更新以強化其反映風險的能力。
- 依據原則 12，監理機關必須發布定義為狀況異常銀行的標準。對於「狀況異常銀行」的定義門檻亦被緊縮，監理機關採用的狀況異常/重要性檢測至少應包括在一系列規定的利率震盪情境下，銀行的 ΔEVE 與其第一類資本 15% 作比較。監理機關可採用自己的具體措施實施額外的狀況異常/重要性測試。

十一、QuantLib 程式庫

(一) QuantLib 程式庫的由來



◆ QuantLib 是一個開放源碼軟體程式庫(Open-Source Software Library)。

- 其開發目的在提供對金融工具評價和相關主題有興趣的軟體開發人員，適合的開發工具。
- 可在 QuantLib 的網站上，www.quantlib.org，了解其專案的相關內容。
 - ✓ QuantLib 是以 C++ 語言開發的程式庫，但被轉寫成其他不同的語言。
 - ✓ 包括 C#、Objective Caml、Java、Perl、Python、GNU R、Ruby 與 Scheme。
- 在 QuantLibAddin/QuantLibXL 專案中，將 QuantLib 的程式庫包裝成 Excel 可以使用的增益集
 - ✓ 一般的財務人員也可透過 Excel 的使用，分享 QuantLib 專案的成果。

◆ QuantLib 的歷史

- 由一群在 Cabota Banca Intesa 的利率衍生商品交易台(Interest Rate Derivative Desk)工作的數量分析專家們，於西元 2000 年時所開創的一個財務金融程式庫開發專案。
 - ✓ 目前這些專家們已經成立了一家公司，該公司之前稱之為 RiskMap，現在則命名為 StatPro Italia。QuantLib 專案目前是由 Luigi Ballabio 與 Ferdinando Ametrano 兩位專家所領導。
 - ✓ QuantLib 所提供的工具，不論對實務上的實作或是進階的建模，都是相當有用的。目前最新的版本是 1.1 版。
- 除了 C++ 語言之外，其他語言的 QuantLib 專案也陸續成立運作，
 - ✓ 例如使用 C# 語言的 QLNet，<http://sourceforge.net/projects/qlnet/>
 - ✓ 使用 R 語言的 RQuantLib，<http://dirk.eddelbuettel.com/code/rquantlib.html>
 - ✓ 使用 Java 語言的 JQuantLib，<http://www.jquantlib.org>
 - ✓ 以及可配合 Excel 使用的增益集，本專案的子專案 QuantLibXL

◆ QuantLib 各版本的演進與推出時間

Version	Release date	Notes
0.1.1	Nov 21, 2000	
0.2.0	Sep 18, 2001	
0.3.4	Nov 21, 2003.	
0.3.7	Jul 23, 2004.	此版之後 QuantLib 需要 Boost 程式庫。
0.4.0	Feb 20, 2007.	
0.8.0	May 30, 2007.	版本的大幅更動，目的在加快收斂到 1.0 版。
0.9.0	Dec 24, 2007.	
0.9.9	Nov 2009.	
1.0.0	Feb 24, 2010	
1.0.1	Sep 17, 2010	
1.1.0	May 23, 2011	

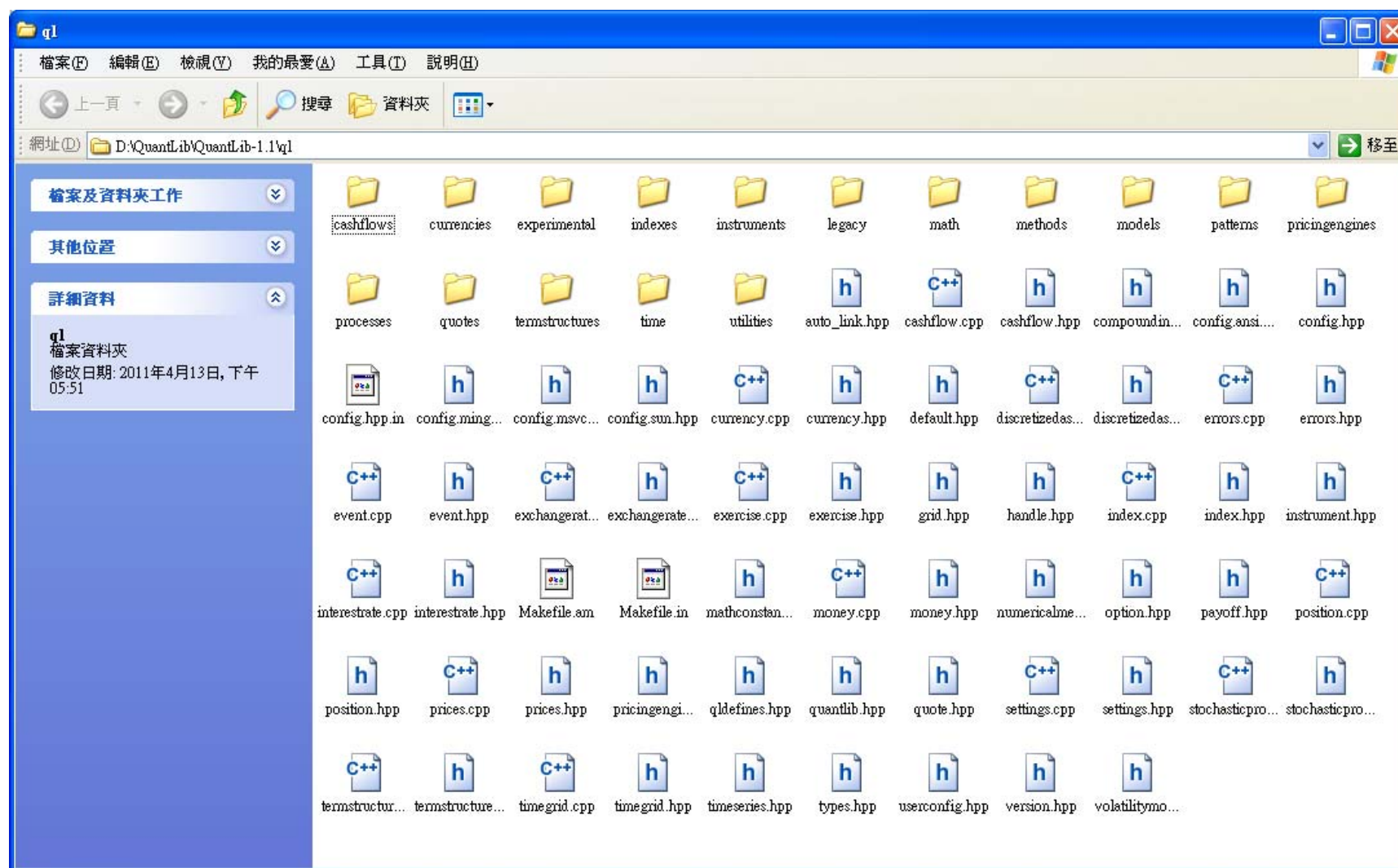
(二)QuantLib程式庫的內容

◆ QuantLib 程式庫內容豐富，包含的檔案上千個，可以概分 15 類模組，如下所示。

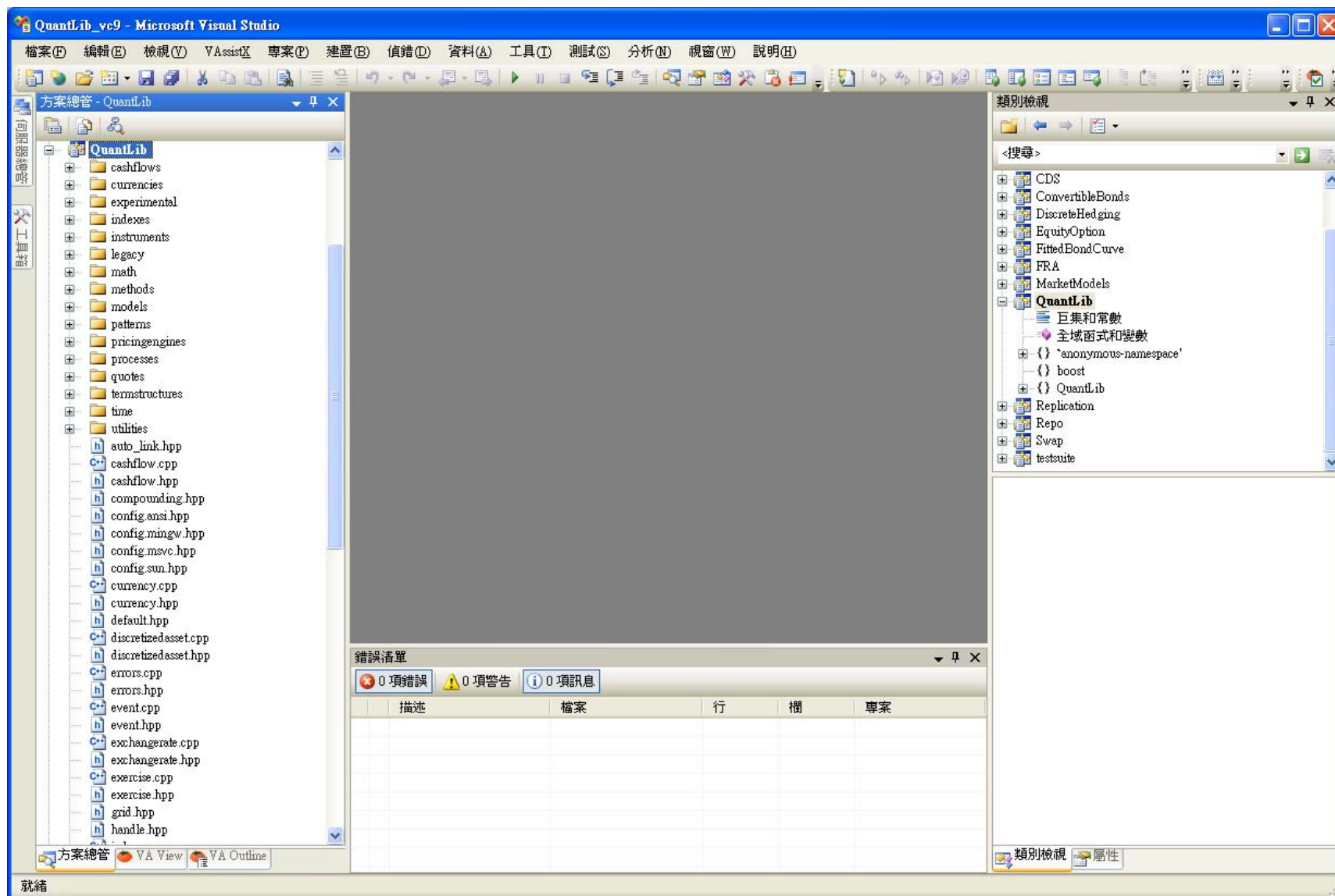
- Numeric types: 主要在定義各類資料的資料型別，例如，利率(Rate)、利差(Spread)、波動性(Volatility) 是實數(Real)。
- Currencies and FX rates：包括 66 種貨幣類別以及相關匯率轉換計算與管理的物件。
- Date and time calculations：包括 37 種不同國家/地區日曆類別、7 種不同計息方式類別以及相關日期計算的物件。
- Pricing engines：九大類評價引擎，包括 Asian option、Barrier option、Basket option、Cap/Floor、Cliquet option、Forward option、Quanto option、Swaption、Vanilla option 等。每一類都有解析解、二元樹、有限差分法與蒙地卡羅模擬的實作類別。
- Finite-differences framework：三大類有限差分法的實作類別。
- Short-rate modelling framework：包括單因子模型，Vasicek、CIR、Hull-White、Black-Karasinski、Extended CIR 等模型。雙因子模型的 G2 模型。

- Financial instruments：40 多種的金融工具，含 Swap、Vanilla Option、Exotic Option、Stock、Forward、Cap、Floor、Color、Bond、Future、Callable Bond 以及相對應的 Quanto 產品、Inflation Bond。
- Lattice methods：一維與二維的二元樹和三元樹模型。
- Math tools：包括分配、積分、相關性、內差、矩陣、最適化、亂數、求解、統計等九類數學工具。
- Monte Carlo framework：單變數與多變數的歐式與美式模擬方法物件。
- Design patterns：Singleton、Observer/Observable、Lazy Object、Composite、Curiously Recurring Template、Acyclic Visitor 樣式。
- Stochastic processes：幾何布朗運動、隨機波動模型、Square Root Process、Ornstein Uhlenbeck Process、Hull White Process、G2 Process。
- Term structures：包括利率、波動性、信用與通膨的期限結構物件。
- Models：分別包括 Equity Model、Short Rate Model、Volatility Model 中使用的相關模型，以及針對 Libor/Swap Market Model 涉及的金融工具與模擬方法的相關物件上百個，。
- QuantLib macros：一些數值極限與除錯所需的 Macros。

◆ 安裝後的目錄內容



◆ QuantLib 專案的大致輪廓



◆ C# Example Code

```
class EquityOption
{
    static void Main(string[] args)
    {
        DateTime timer = DateTime.Now;

        // set up dates
        Calendar calendar = new TARGET();
        Date todaysDate = new Date(15, Month.May, 1998);
        Date settlementDate = new Date(17, Month.May, 1998);
        Settings.setEvaluationDate(todaysDate);

        // our options
        Option.Type type = Option.Type.Put;
        double underlying = 36;
        double strike = 40;
        double dividendYield = 0.00;
        double riskFreeRate = 0.06;
        double volatility = 0.20;
        Date maturity = new Date(17, Month.May, 1999);
        DayCounter dayCounter = new Actual365Fixed();
        Console.WriteLine("Option type = " + type);
    }
}
```

```

Console.WriteLine("Maturity = " + maturity);
Console.WriteLine("Underlying price = " + underlying);
Console.WriteLine("Strike = " + strike);
Console.WriteLine("Risk-free interest rate = {0:0.000000%}", riskFreeRate);
Console.WriteLine("Dividend yield = {0:0.000000%}", dividendYield);
Console.WriteLine("Volatility = {0:0.000000%}", volatility);
Console.WriteLine("\n");
string method;
Console.WriteLine("\n");

// write column headings
int[] widths = new int[] { 35, 14, 14, 14 };
Console.WriteLine("{0,-" + widths[0] + "}", "Method");
Console.WriteLine("{0,-" + widths[1] + "}", "European");
Console.WriteLine("{0,-" + widths[2] + "}", "Bermudan");
Console.WriteLine("{0,-" + widths[3] + "}", "American");
List<Date> exerciseDates = new List<Date>(); ;
for (int i = 1; i <= 4; i++)
    exerciseDates.Add(settlementDate + new Period(3 * i, TimeUnit.Months));
Exercise europeanExercise = new EuropeanExercise(maturity);
Exercise bermudanExercise = new BermudanExercise(exerciseDates);
Exercise americanExercise = new AmericanExercise(settlementDate, maturity);
Handle<Quote> underlyingH = new Handle<Quote>(new SimpleQuote(underlying));

```

```

// bootstrap the yield/dividend/vol curves
var flatTermStructure = new Handle<YieldTermStructure>(new FlatForward(settlementDate, riskFreeRate, dayCounter));
var flatDividendTS = new Handle<YieldTermStructure>(new FlatForward(settlementDate, dividendYield, dayCounter));
var flatVolITS = new Handle<BlackVolTermStructure>(new BlackConstantVol(settlementDate, calendar, volatility, dayCounter));
StrikedTypePayoff payoff = new PlainVanillaPayoff(type, strike);
var bsmProcess = new BlackScholesMertonProcess(underlyingH, flatDividendTS, flatTermStructure, flatVolITS);

// options
VanillaOption europeanOption = new VanillaOption(payoff, europeanExercise);
VanillaOption bermudanOption = new VanillaOption(payoff, bermudanExercise);
VanillaOption americanOption = new VanillaOption(payoff, americanExercise);

// Analytic formulas:
method = "Jarrow-Rudd Tree";
europeanOption.setPricingEngine(new BinomialVanillaEngine<JarrowRudd>(bsmProcess, 12));
Console.WriteLine("{0,-" + widths[0] + "}", method);
Console.WriteLine("{0,-" + widths[1] + ":0.000000}", europeanOption.NPV());
Console.WriteLine("{0,-" + widths[2] + "}", "N/A");
Console.WriteLine("{0,-" + widths[3] + "}\n\n", "N/A");

// Black-Scholes for European
method = "Black-Scholes";
europeanOption.setPricingEngine(new AnalyticEuropeanEngine(bsmProcess));
Console.WriteLine("{0,-" + widths[0] + "}", method);

```

```

Console.Write("{0,-" + widths[1] + ":0.000000}", europeanOption.NPV());
Console.Write("{0,-" + widths[2] + "}", "N/A");
Console.WriteLine("{0,-" + widths[3] + "}", "N/A");

// Barone-Adesi and Whaley approximation for American
method = "Barone-Adesi/Whaley";
americanOption.setPricingEngine(new BaroneAdesiWhaleyApproximationEngine(bsmProcess));
Console.Write("{0,-" + widths[0] + "}", method);
Console.Write("{0,-" + widths[1] + "}", "N/A");
Console.Write("{0,-" + widths[2] + "}", "N/A");
Console.WriteLine("{0,-" + widths[3] + ":0.000000}", americanOption.NPV());

// Bjerk Sund and Stensland approximation for American
method = "Bjerk Sund/Stensland";
americanOption.setPricingEngine(new Bjerk SundStenslandApproximationEngine(bsmProcess));
Console.Write("{0,-" + widths[0] + "}", method);
Console.Write("{0,-" + widths[1] + "}", "N/A");
Console.Write("{0,-" + widths[2] + "}", "N/A");
Console.WriteLine("{0,-" + widths[3] + ":0.000000}", americanOption.NPV());
}
}

```