

## 一、YTM 計算

透過債券現值的公式

$$PV = C \frac{1 - (1 + r)^{-n}}{r} + \frac{F}{(1 + r)^n}$$

可以知道當我們有債券現值、票面利率、面額和期長就可以透過解  $r$  的一元高次方程式算出對應的殖利率，這部分的程式主要是透過二分查找解。這裡有個比較特別的地方，由於網頁的計算器在年份有小數、期長為 **semi** 或 **quarter** 的時候我不知道它背後程式的算法。所以我自己做了一個這種情況下的 YTM，舉個例子：若年數為 2.4，半年繳一次，那麼總期數就是  $2.4 \times 2 = 4.8$ ，我判定只會拿到 4 期的票息分別在 1、2、3、4 期，在 4.8 期拿到面額，但這種方式做出來的殖利率和網頁計算的不同，我不確定它是會直接捨棄後面那 0.8 期或者是第一筆錢不是在第一期時(可能在 0.8、1.8...)拿到，但我可以確保程式應該沒有錯誤。

```
Bond price(元): 950
Par Value(元): 1000
Coupon Rate(%): 3.245
Years to Maturity(年): 4.7
Payment(A,S,Q): s
0.0224218
Process returned 0 (0x0)    execution time : 12.455 s
Press any key to continue.
```

前五行會詢問資料，而第 6 行就會給出殖利率，這邊有一點要注意一下：他算出的殖利率其實是期的，而不是年利率，我會這樣做是為了和網頁上的計算器做對比(因為網頁上的跑出來也是期利率)，如這邊的  $ytm=0.0224218$ ，實際換算成年利率應該要乘 2(semi)。

## 二、即期利率和遠期利率計算

即期利率由於網頁上的是給零息債券，所以只要帶入債券售價、面額、期長

$$PV = \frac{F}{(1 + r)^n}$$
$$r = \left( \frac{F}{PV} \right)^{\frac{1}{n}} - 1$$

就可以求出對應的 **spot rate**。

程式中，**spot rate** 的計算是一個一行的函數(寫成函數版面比較乾淨)

因此，我們只要輸入多張不同時長的債券資料(市值、面額、期長)就可以得到對應的 **spot rates**。

$$f(i, j) = \left( \frac{(1 + S_j)^j}{(1 + S_i)^i} \right)^{j-i} - 1$$

透過上面這個公式和各種即期利率我們就可以推出不同期間的遠期利率。

```

請輸入一組債券市價: 900 800 700 600 500 400 300 200
請輸入對應面額: 1000 1000 1000 1000 1000 1000 1000 1000
請輸入對應期長: 1 2 3 4 5 6 7 8
S1=0.111111 S2=0.118034 S3=0.126248 S4=0.136219 S5=0.148698 S6=0.164993 S7=0.187673 S8=0.222845
第1期 第2期 第3期 第4期 第5期 第6期 第7期 第8期
第1期 0 0.1250 0.1339 0.1447 0.1583 0.1761 0.2009 0.2397
第2期 X 0 0.1429 0.1547 0.1696 0.1892 0.2167 0.2599
第3期 X X 0 0.1667 0.1832 0.2051 0.2359 0.2847
第4期 X X X 0 0.2000 0.2247 0.2599 0.3161
第5期 X X X X 0 0.2500 0.2910 0.3572
第6期 X X X X X 0 0.3333 0.4142
第7期 X X X X X X 0 0.5000
第8期 X X X X X X X 0
Process returned 0 (0x0) execution time : 39.563 s
Press any key to continue.

```

資料的輸入在前三行，基本上可以輸入任意多組(用 **deque** 動態記憶)，做成任意多組的原因是如果今天可以在網路上把債券的資料都載下來存進 **excel** 檔案，那麼 **C++** 只要用 **csv** 讀取就可以跑表格，處理上會比一個一個輸入好，也才能處理大數據。而對應期長則是每張債券的分期，沒有直接寫 **8** 而要輸入 **1~8** 是因為可能會有 **1.5,2.3,3.6** 這類的小數期長。