**財務演算法 作業5 Monte Carlo Simulation**

資工四 B96902041 吳承恩

**執行方法:**

使用DevC++ 4.992開啟hw5.cpp後直接compile並執行

接著依照提示輸入S1 , S2 ,S3 ,X ,H ,year ,δ1 ,δ2 ,δ3 ,ρ ,rate ,n ,m

然後就會執行五種方法來算選擇權的Price以及delta和Gamma

Method1 : Origin method 原始的蒙地卡羅法

Method2 : Discretize dS/S

Method3 : Brownian Bridge Approach

Method4 : Method1 and Variance reduction

Method5 : Method3 and Variance reduction

**撰寫語言:**

C++

**細部說明:**

這次作業我嘗試了各種方法比去作比較

分別是

**Brownian Bridge Approach:**

利用布朗運動的特性去預估路徑是否曾碰觸到barrier，若最後價值超過barrier，就代表一定碰到過，若最後價值不超過barrier，則沒碰到的機率: 1-exp(-(2ln(H/S)\*ln(H/S(T)) )/ (δ^2\*T) ) ，利用此方式來評估價值，此方法最大的好處是同樣的replication數(N)時，不需要將時間切割成許多小片段來計算，因此計算速度較快，時間複雜度為O(N)，而原始方法的時間複雜度為O(N\*M)，M是把時間切割成幾期的期數，我自己測試是發現M至少要大於30才會比較準確，因此布朗橋的方法真的快了不少。

**Variance Reduction:**

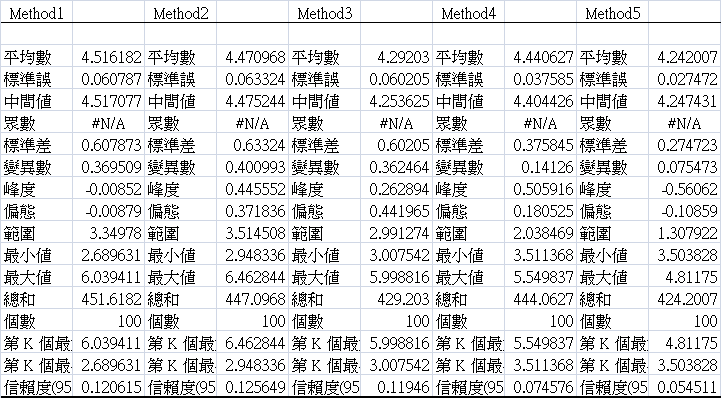
蒙地卡羅法主要是基於大數方則中的中央極限定理，根據統計學，重複計算次數越多平均值就越接近母體的平均值，收斂方程式為O(N^(-1/2)) 也就是說想要降低10倍的誤差就要跑100倍的重複計算次數，variance reduction指的就是如何在減少誤差而又不需要多跑那麼多次，我實作的方式是Antithetic Variates，其原理就是利用Var((Y1+Y2)/2) = Var(Y1)/2+Cov(Y1,Y2)/2

Y1跟Y2是同樣的distribution(在這次作業中同樣都是Normal distribution) 而Y2為負的Y 1，故Cov(Y1,Y2)=-1，因此這方法降低了1/2的variance，各多作一次負的試驗就大幅降低了variance。

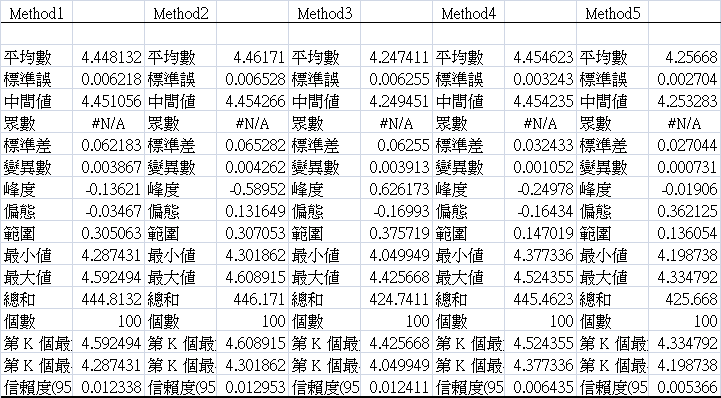
**實驗數據:**

這次作業我用了excel來稍微統計一下各種方式的結果

表一: n=100 m=50 每個方式的100筆結果



表二: n=10000 m=50每個方式的100筆結果



Variable reduction的效用可以從表一跟表二中的method3跟method5及method1跟method4的變異數比較中明顯的看出

而Brownian bridge approach的效用就是比較method1跟method3 的變異數， 兩表中的變異數都相差不大，然而method1的計算時間是method3的(m=50)倍，故Brownian bridge approach的收斂速度較快。

**檔案說明:**

B96902041\_HW\_5.zip中內含四個檔案

Hw5.cpp 程式原始碼

Hw5.exe 程式執行檔

Book1.xlsx 實驗數據的excel檔，原始數據存到sheet1中，另外兩個sheet則是表一跟表二

Readme.pdf : 即為此說明檔