112 選擇權-評價與應用

期末程式報告要點

1. 提醒事項

I. 此份報告占期末考成績 50%。

II. 繳交方式: **PDF 書面報告及 Python 程式碼**；寄至助教信箱: 111352503@g.nccu.edu.tw

III. 繳交期限: 期末考前一日，即 2024/01/10(週三)。

IV. 報告內容: 原則上依照老師課堂中所述，美式選擇權以三種方法(CRR、BAW、LSM)進行評價並比較三者之間的差異；由於課堂上已附上絕大多數的程式碼，因此若只做以上這部分，報告成績可能只會拿到「基本」的分數，所以可盡量多發想一些有趣的研究議題，細節的部分可自行發揮，並附上實證結果。若實在苦無研究方向，可參考以下推薦的 **Extension works**.

V. 台指選擇權(TXO) market data:

https://www.taifex.com.tw/cht/3/optDailyMarketView

2. Basic requirement

(1) Implement the following option pricing methods to value American-style plain vanilla calls/puts

$$\begin{cases} CRR\ binomial\ tree\ model \\ Efficient\ Analytic\ Approximation(BAW) \\ least\text{-}squares\ Monte\ Carlo\ simulation \end{cases}$$

(Inputs: S_0, K, r, q, σ, T, number of simulations, number of repetitions, n. Outputs: Option values for all methods and 95% confidence interval for least-squares Monte

Carlo simulation.)

Note: 95% confidence interval = [mean of 20 repetitions − 2×(s.d. of 20 repetitions), mean of 20 repetitions + 2×(s.d. of 20 repetitions)].

[bonus] Many papers improve the LSM by using more advanced regression methods, such as ridge regression (Tompaidis and Yang, 2014), least absolute shrinkage and selection operator (LASSO) (Tompaidis and Yang, 2014; Chen et al., 2019), weighted least squares regression (Fabozzi et al., 2017; Ibanez and Velasco, 2018), and non-parametric kernel regression (Belomestny, 2011; Ludkovski, 2018). **Please attempt to replace the Ordinary Least Squares (OLS) regression in Least-Squares Monte Carlo with the advanced regression methods mentioned above and comparing their differences.**

3. Extension works (optional)

(1) Calculate the implied volatility with

- Black-Scholes model
  - European calls
  - European puts
- Binomial tree model
  - European calls
  - European puts
  - American calls
  - American puts

+

- Bisection method
- Newton's method

(Inputs: S_0, K, r, q, T, market price of the option, n, convergence criterion. Outputs: Implied volatility.) Show your empirical results to substantiate the phenomenon of a volatility smirk in equity options (including index options).

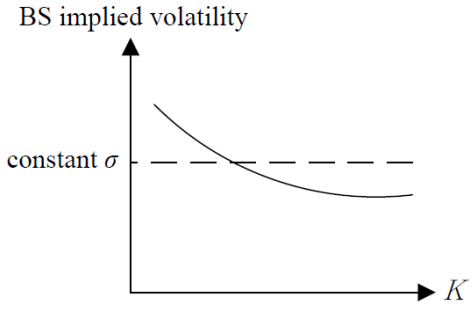
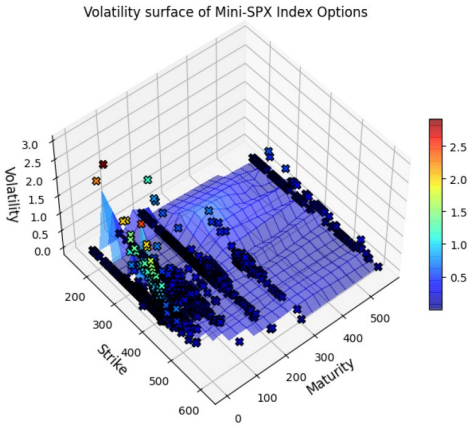


Figure: Illustration of the Volatility Smirk for Equity Options

(2) Plot the volatility surface (x: strike, y: maturity, z: implied volatility) of options issued on the index or equity options. Offer interpretations for the empirical results obtained.

An example of volatility surface of S&P 500 Mini SPX Options Index is given by:



(3) Implement the implicit and explicit finite difference methods to price American and European calls and puts. (Inputs: S_0, K, r, q, σ, T, $S_{\min}$, $S_{\max}$, m (number of partitions for the stock price), n (number of partitions for the time to maturity). Outputs: Option values.)

(4) BAW 的實證結論: 「對於到期日短於一年的美式選擇權，BAW 的評價模型不但準確，其速度也極快(相對於有限差分法)。若到期日一年以上，建議以有限差分法求解美式選擇權，因 BAW 方法的準確度不如前兩者的準確度。」**你同意上述的實證結果嗎? 請使用實際市場資料驗證其結果是否符合**。

(5) 在蒙地卡羅模擬法中，亂數的產生是一個重要的課題，程式中，需要亂數時，通常可以由內建程式庫中取得亂數。這類亂數可稱之為假亂數(Pseudo-Random Number)，這是因為我們是採用確定的方法(Deterministic Method)去模仿產生亂數，本質上他們並不是真正的亂數，只是刻意地使其看起來像是亂數而已。不過，它們確實具備了一些亂數該有的性質。另一類在程式設計中可充當亂數來源的是所謂的準亂數(Quasi-Random Number)，這類亂數在本質上就不是以模仿亂數的方式去產生，他們是以數論的理論去產生均勻分佈的數列(Low-discrepancy Sequence)。

事實上，如果要產生「真正」的亂數，那還是要求助於大自然，原子核衰變過程中，放射出來的粒子的方位，便是一個天然的亂數源。網路上

有網站提供(氪-85)[1]產生的數列，這是一個有趣的網站。http://www.fourmilab.ch/hotbits/；此外，Toshiba 使用半導體上的熱騷動(Thermal Noise)，以 PCI 板方式販售 Random Master，可相容用於 PC 與超級電腦，是利用了磁碟機轉動產生的空氣亂流(Air Turbulence)。

一個良好的亂數產生器，應該具備以下三項性質，產出的亂數要能均勻分佈、亂數數列的週期要夠長、產生的速度要夠快。其中，在均勻分布方面，理論上的均等分佈亂數，要能均勻的分佈於範圍之內，對於一維的亂數，這應該不是太大的問題。然而，如果我們要的是 100 維的亂數，則演算法的品質就可能有差異了。一些演算法在高維度時，會出現群聚(Clustering)的現象，造成高維空間分佈不均勻的情況。實務上，一個在過去 20 年頗受歡迎的演算法，已逐漸為各個主要數值程式庫所採用。日本學者 Makoto Matsumoto (松本真)與 Takuji Nishimura (西村拓士)於 1997 年發表的 Mersenne Twister (MT)亂數產生器(MT19937)。在有限二進位位元上引入線性遞歸技術，解決了很多傳統 LCG 的問題，可以快速產生高品質的假亂數。不過，MT19937 仍存在一些缺點，在 Numpy 的 document 中，可以看到以下說明:

Random sampling (`numpy.random`)

NumPy implements several different `BitGenerator` classes implementing different RNG algorithms. `default_rng` currently uses `PCG64` as the default `BitGenerator`. It has better statistical properties and performance than the `MT19937` algorithm used in the legacy `RandomState`. See Bit Generators for more details on the supported BitGenerators.

亂數產生器 default 使用 PCG64，具備一些良好的統計性質，優於 MT19937。

以上蒙地卡羅模擬法的隨機亂數產生，乃基於 pseudo random sequence 的方法所產生；不過在定積分的理論中，則常常使用準亂數，又稱之為低差異數列(Low-discrepancy Sequence, LDS)，此數列中的點，落於特定區域 B 的比率，與該區域 B 的空間測度成正比。這些序列所產生的樣本完全不具任何隨機性，並且相當規律的以非常均勻的方式分散在樣本空間中，使用 LDS 來取代蒙地卡羅模擬法中的隨機亂數，就稱為準蒙地卡羅模擬法，能有效提高收斂速度。其中一類 LDS 族群為 digital net sequence，例如: Halton 序列、Sobol 序列。不過，值得一提的是，LDS 數列在高維度時會發生叢聚的現象，使得亂數分布產生不均勻的結果；此外，相對 Halton 數列，Sobol 數列的品質較佳(至 256 維皆可均勻分布)。

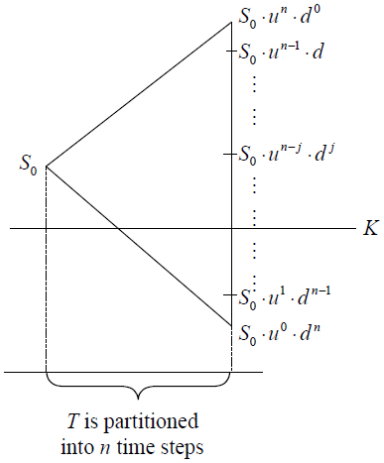
[1] 惰性氣體: 氦（He）、氖（Ne）、氬（Ar）、氪（Kr）、氙（Xe）和具放射性的氡（Rn）；氪-85 (氪 85)是氪的一種放射性同位素。

有了以上隨機亂數的基本概念後，請嘗試使用準蒙地卡羅模擬法，即亂數的部分使用 Low-discrepancy Sequence (LDS)，並與傳統的蒙地卡羅模擬法進行比較。

(6) Implement the **combinatorial method** to price European calls and puts.

- Combinatorics (組合數學) is a branch of pure mathematics concerning the study of discrete (and usually finite) objects. Aspects of combinatorics include “counting" the objects satisfying certain criteria, deciding when the criteria can be met, or finding "largest", "smallest", or "optimal" objects.
- Combinatorial method for European options

  Based on the binomial tree framework, applying the combinatorial method is far faster than the backward induction procedure. In fact, it is not necessary to build the binomial tree in the combinatorial method, which is another advantage of the combinatorial method because it can save memory space for computer programming.



European option value $= e^{-rT} \sum_{j=0}^{n} \binom{n}{j} p^{n-j}(1-p)^j \max(S_0 u^{n-j} d^j - K, 0)$, where $\binom{n}{j}$, also denoted as C_j^n, is the combination of j from n, $u = e^{\sigma\sqrt{\Delta t}}$, $d = e^{-\sigma\sqrt{\Delta t}}$, and $p = \frac{e^{(r-q)\Delta t} - d}{u-d}$.

以上是 6 個推薦的 extension researches，但切記不要只把程式跑出來後展示圖表結果，須有充分的文字論述、研究發現與經濟意涵。