

# ASIAN OPTIONS MONTE CARLO PRICING USING THE LÉVY LOGNORMAL APPROXIMATION

THEO DIMITRASOPOULOS\*, WILLIAM KRAEMER\*, SNEHAL RAJGURU\*\*, VAIKUNTH SESADRI\*

\* Department of Financial Engineering; Stevens Institute of Technology Babbio School of Business

\*\* Department of Engineering Management; Stevens Institute of Technology School of Systems & Enterprises

**KEYWORDS**— Arithmetic Average, Asian Options, Brownian Motion, Control Variates, Derivatives Pricing, Geometric Average, Geometric Brownian Motion, Lévy Approximation, Lognormal Approximation, Monte Carlo Simulation, Options Pricing, Python, Statistical Modeling, Variance Reduction

## I. INTRODUCTION

Asian Options are an exotic derivative where the payoff is determined by the average price of the underlying asset over its maturity T i.e. path dependent contract. The payoffs of a fixed-strike Asian option at maturity T are defined as,

$$C(K, T) = (K - \bar{A}_T, 0)^{\wedge} + \quad 1$$

$$P(K, T) = (\bar{A}_T - K, 0)^{\wedge} + \quad 2$$

Averaging can be in continuous/discrete time

$$\bar{A}_T = \frac{\int_0^T S_t dt}{T} \text{ i.e. the average price of the underlying asset.}$$

K = Strike price      (this is continuous-time average)

The average can be calculated arithmetically or geometrically, a detail that is explicitly stipulated in the option contract. They are defined as, (this is discrete-time average)

$$A_N = \frac{1}{N} \sum_{i=1}^N S_{t_i} \quad 3$$

$$G_N = (\prod_{i=1}^N S_{t_i})^{\wedge(1/N)} \quad 4$$

$S_{t_i}$  = Underlying asset prices at times  $t_i$  for  $i = 1, 2, \dots, N$

Asian options using arithmetic averaging procedure are more common than those using geometric averaging. However, the advantages of a geometric average procedure make it a very appealing choice for quantitative modeling and will be discussed further in the following sections.

Most Asian option contracts in industry are known to be averaging-in style. Averaging-in style contracts will sample at discrete and regular time intervals i.e weekly, monthly, quarterly, or annually averaging from inception to maturity date. Conversely, if an Asian option is averaging-out style the average is computed using specific samples near the maturity date. Generally speaking, averaging-out style options are more risky than averaging-in style options since the uncertainty of future spot prices is higher the further in time from the start of the contract. Averaging-out style Asian options have an increased risk exposure to spot price and volatility which is reflected in their higher price.

Arithmetic Asian option prices can only be estimated using numerical methods such as Monte Carlo, because the arithmetic average of lognormal variables do not follow a lognormal distribution. However, an analytical closed form formula for Geometric Asian Options can be adapted from the Black-Scholes-Merton formula, since geometric averages of log-normal variables also follow a lognormal distribution in a risk neutral world. Kemna and Vorst offer a good approximation for an geometric Asian option using the Black-Scholes formula which will be discussed in the pricing algorithm section.[reference to KV paper [2]]

Compared to similar ‘vanilla’ exchange traded derivatives Asian options are attractive as their volatility is low due to the averaging mechanics of its payout. Consequently, Asian options are less expensive than their corresponding vanilla derivatives. Furthermore, the higher number of observations, the lower the price of the Asian option. For this reason Asian options are a favored hedging tool of actors in volatile markets such as commodities, foreign exchange, and energy. Consider an airline that steadily buys crude oil whose supply price is not fixed, but set weekly from a particular benchmark. The airline may hedge themselves against a spike in oil by using a tailored Asian option

to reflect the weekly purchases. This would be less expensive and more convenient than buying a basket of European options expiring at weekly intervals. Asian options are also useful in markets with low volume, as the derivative would protect against price manipulation of the underlying asset.

## II. MATERIALS AND METHODS

### A. Geometric Brownian Motion aka Black-Scholes model

The underlying asset price is assumed to follow a stochastic process of a Geometric Brownian Motion (hereinafter GBM),

$$S = rSdt + \sigma Sdz \quad 5$$

$S$  = price of the underlying asset

$r$  = risk-free interest rate

$\sigma$  = volatility

$dz$  = Wiener process

A Wiener process has the following properties:

1. The change  $\Delta z$  in a short period of time  $\Delta t$  is,

$$\Delta z = \varepsilon\sqrt{\Delta t} \quad 6$$

where  $\varepsilon \sim N(0,1)$

2. The values of  $\Delta z$  for any two different short intervals of time  $\Delta t$  are independent.

$N(x, y)$  denotes a normal distribution with mean  $x$  and variance  $y$ . In discrete time terms, the change in the stock price from Eq. 5 & 6 becomes,

$$\begin{aligned} \Delta S &= rS\Delta t + \sigma S\varepsilon\sqrt{\Delta t} \\ \frac{\Delta S}{S} &= r\Delta t + \sigma\varepsilon\sqrt{\Delta t} \\ \frac{\Delta S}{S} &\sim N(r\Delta t, \sigma^2\Delta t) \end{aligned} \quad 7$$

Therefore, the rate of return of the stock price over a time interval  $\Delta t$  follows a normal distribution with mean  $r\Delta t$ , and standard deviation  $\sigma\sqrt{\Delta t}$ . Applying Itô's lemma, if a stochastic variable  $X$  follows the Itô process,

$$dX = a(X, t)dt + b(X, t)dz \quad 8$$

A function  $f(X, t)$  follows the process,

$$df = \left( \frac{\partial f}{\partial X} a + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial X^2} b^2 \right) dt + \frac{\partial f}{\partial X} bdz \quad 9$$

This can in turn be applied to the stock price process in Eq. 5 to define the continuous path,

$$\begin{aligned} dln(S) &= \left( \frac{1}{2}rS + 0 + \frac{1}{2}\left(\frac{-1}{S^2}\right)\sigma^2 S^2 \right) dt + \frac{1}{2}\sigma S dz \\ &= \left( r - \frac{1}{2}\sigma^2 \right) dt + \sigma dz \end{aligned} \quad 10$$

The equivalent of Eq. 10 in discrete time is therefore,

$$\begin{aligned} \Delta \ln(S) &= \left( r - \frac{1}{2}\sigma^2 \right) \Delta t + \sigma\varepsilon\sqrt{\Delta t} \Leftrightarrow \\ \ln(S_{t+\Delta t}) - \ln(S_t) &= \left( r - \frac{1}{2}\sigma^2 \right) \Delta t + \sigma\varepsilon\sqrt{\Delta t} \Leftrightarrow \\ S_{t+\Delta t} &= S_t * e^{\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}} \end{aligned} \quad 11$$

Eq. 9 is the GBM price path generator that is used to construct hypothetical trajectories of the price of the underlying asset. The following pseudocode describes the constructor,

**function** GBM( $S_0, K, T, r, \sigma, n_{\text{paths}}, n_{\text{steps}}$ ):

Define a timestep  $\Delta t$  by dividing the maturity ( $T$ ) input by the desired number of discrete steps (price values at time  $t_i$  along the path,  $n_{\text{steps}}$ )

Generate an array  $\varepsilon$  of size  $n_{\text{paths}}$  rows and  $n_{\text{steps}}$  columns, where each column represents one price path.

Fill the array with values drawn from a standard normal distribution i.e.  $\varepsilon \sim N(0,1)$ .

Perform the operation described in Eq. 9 as a cumulative product of the exponents of the random sample with

$$\text{Mean} = \left( r - \frac{1}{2}\sigma^2 \right) \Delta t \quad 12$$

$$\text{Variance} = \sigma * \varepsilon\sqrt{\Delta t} \quad 13$$

Fig. 1. GBM price path generator definition.

The mean and variance of the price path are the risk-free interest rate and the stock volatility of the underlying asset respectively.

GBM is the mathematical concept, and the Black-Scholes model is the financial model for the asset price

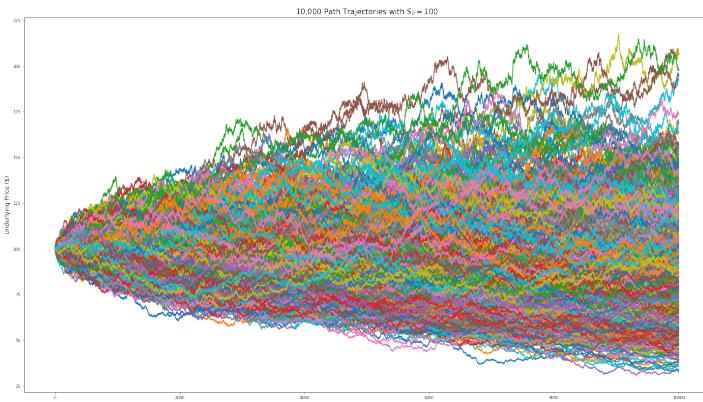


Fig. 2. Result for  $n_{\text{paths}} = 1,000$  price paths with  $n_{\text{steps}} = 10,000$  timesteps, for an underlying with initial price  $S_0 = \$100$ .

Fig. 2 shows the result of  $n_{\text{paths}} = 1,000$  price paths with  $n_{\text{steps}} = 10,000$  timesteps.

### B. Lognormal (Lévy) approximation and $A_N$ :

The valuation of Arithmetic average Asian options under standard assumptions poses various difficulties. Central to the valuation issue is, with the exception of trivial cases, the lack of closed-form solutions under the conventional geometric diffusion model for the underlying price process. In essence, with no The Lévy approximation [1] solves the issue and enables an accurate approximation of the value of arithmetic average Asian options. More specifically, the hypothesis that “the distribution of an arithmetic average is well-approximated by the lognormal distribution when the underlying price process follows the conventional assumption of a geometric diffusion” [1], is confirmed with tests that simulate volatility similar to real market conditions. This fundamental assumption forms the basis of a performance benchmark of the pricing algorithm implementation against the standard Linetsky test cases [2].

The lognormal approximation arises from the moment matching procedure described in the Appendix wherein the mean ( $M_1$ ) and variance ( $M_2$ ) of  $\int_0^T S_t dt$  in  $\mathcal{P}^*$  is,

$$\mu_{\text{lognormal}} = \frac{(e^{rT} - 1)}{rT} S_0$$

$$\sigma_{\text{lognormal}} = \frac{2S_0^2}{T^2(r + \sigma^2)} \left( \frac{e^{2r + \sigma^2} T - 1}{2r + \sigma^2} - \frac{e^{rT} - 1}{r} \right) - \left( \frac{e^{rT} - 1}{rT} S_0 \right)$$

Once we make the log-normal assumption, we can price the Asian option analytically. No need for MC simulation. The MC simulation based on GBM paths goes beyond the log-normal assumption.

$$\ln(X) \sim \mathcal{N}(\mu_{\text{lognormal}}, \sigma_{\text{lognormal}}^2).$$

Linetsky does not use the log-normal approximation. Also the MC simulation does not make this assumption. The Levy (log-normal) method is a useful quick (but not very precise) approximation. See the attached map of simulation methods.

### Black-Scholes Merton Formula for Asian Options with $G_T$

$$G_C(K, T) = e^{-rT} * [G_0 * N(d_1) - K * N(d_2)] \quad 14$$

$$G_P(K, T) = e^{-rT} * [K * N(-d_2) - G_0 * N(-d_1)] \quad 15$$

$$d_{1,2} = \frac{1}{\Sigma_G \sqrt{T}} * (\log \frac{G_0}{K} \pm \frac{1}{2} \Sigma_G^2 * T) \quad 16$$

$$\Sigma_G = \frac{1}{\sqrt{3}} \sigma \quad 17$$

$$G_0 = S_0 * e^{\frac{1}{2}(r-q)T - \frac{1}{12}\sigma^2 T} \quad 18$$

Fig. 3. Modified Black-Scholes-Merton Formula for Geometric Average Asian Options.

### C. The modified Black-Scholes-Merton Model:

In the case of Geometric Average Asian Options, the Black-Scholes-Merton options pricing model (hereinafter referred to as BSM model) is fully compatible and can be solved in closed-form. This arises from the inherent lognormality of the two distributions. Therefore, the modified BSM PDE for geometric average call and put options is defined as shown in Fig. 3. why PDE? These are the analytical solutions for the geometric average Asian options in the BS model

The modified volatility  $\Sigma_G$  allows for the treatment of Asian options as European options with a decreased volatility by a factor of  $\sqrt{3}$ . Under the Lévy approximation [1], the accuracy of both arithmetic and geometric average Asian options can be measured through direct comparisons with the closed form BSM model. This confirms the shared lognormality of the geometric average and enables additional stress test against arithmetic average contracts, which are more commonly used in industry.

### D. Monte Carlo Simulator

Monte Carlo experiments (hereinafter referred to as MC) are a class of computational algorithms that optimize a numerical result of a deterministic problem through the use of repeated random sampling. In summary, the following steps are executed in an MC simulation,

- Define the input domain for the task,
- Generate random inputs from a chosen probability distribution,
- Compute the numerical result for a number of iterations,
- Aggregate the results.

function MC( $S_0$ , K, T, r, q,  $\sigma$ , n<sub>paths</sub>, n<sub>steps</sub>):

Define an array **PAYOUTS** that will contain the call payoff calculations. t<sub>i</sub> along the path, n<sub>steps</sub>)

Define an array **S** that will contain the price paths from GBM

**for** i 1 : simulations

**S** = results of GBM( $S_0$ , K, T, r,  $\sigma$ , n<sub>paths</sub>, n<sub>steps</sub>)

Average across each column, either arithmetically or geometrically (separate definitions), using **Eq. 3 or 4**

Calculate the discounted payoffs of the options in the vector of average prices using **Eq. 1 or 2** and discount factor  $e^{-rT}$

Average across the **PAYOUTS** from the previous steps

Append the discounted payoff average of simulation i in **PAYOUTS**

At the end of the loop, calculate the average of the **PAYOUTS** vector.

Fig. 4. Monte Carlo general algorithm definition.

The GBM Generator implementation creates the aforementioned random sample and computes the price paths. The particular definition in this paper takes the GBM path array, averages the prices along each path using both an arithmetic and geometric mean, calculates the price of the call or put option and aggregates the option prices by taking the average value. An additional property of the definition allows for variance reduction through the use of the geometric average payoff as a control variate for the arithmetic average. The difference of the two averaging methods is then added back to the arithmetic average to generate narrower confidence intervals and better approximate the true mean of the option price. The functions are divided in separate definitions to optimize computational performance, but Fig. 4 presents the general definition for arithmetic and geometric average Asian Call and Put options. Each simulation invokes the GBM function for a given number of iterations and generates an average payoff over the number of simulations. The final result then becomes the average MC price estimate of the option.

```
def mc_call_arithm_tf(enable_greeks=True):
```

```
(S0, K, dt, T, sigma, r, dw, S_i) = tf_graph_gbm_paths()
```

```
A = tf.reduce_sum(S_i, axis=1)/(T/dt) # Arithm. Average
```

```
A = tf.pow(tf.reduce_prod(S_i, axis=1), dt / T) # Geom. Average
```

```
payout = tf.maximum(A - K, 0) # Call Option
```

```
payout = tf.maximum(K - A, 0) # Put Option
```

```
npv = tf.exp(-r * T) * tf.reduce_mean(payout)
```

```
tar = [npv]
```

```
if enable_greeks:
```

```
    greeks = tf.grad(npv, [S0, sigma, dt])
```

```
    dS_2nd = tf.grad(greeks[0], [S0, sigma, dt])
```

```
    dsig_2nd = tf.grad(greeks[1], [S0, sigma, dt])
```

```
    dT_2nd = tf.grad(dS_2nd[0], [S0, sigma, dt])
```

```
    dsig_3rd = tf.grad(dsig_2nd[1], [S0, sigma, dt])
```

```
    tar ± [greeks, dS_2nd, dsig_2nd, dT_2nd, dsig_3rd]
```

```
def pricer(S_zero, strk, maturity, volatility, riskfrate, seed, sim, step):
```

```
if seed != 0:
```

```
    np.random.seed(seed)
```

```
    stdnorm_rand = np.random.randn(sim, step)
```

```
with tf.Session() as sess:
```

```
    delta_t = maturity / step
```

```
    res = sess.run(tar, {
```

```
        S0: S_zero,
```

```
        K: strk,
```

```
        r: riskfrate,
```

```
        sigma: volatility,
```

```
        dt: delta_t,
```

```
        T: maturity,
```

```
        dw: stdnorm_rand})
```

```
return res
```

```
return pricer
```

Fig. 5. Monte Carlo general algorithm definition in Tensorflow.

**n<sub>paths</sub>** = **100** (1 option price  $C(K, T)$  per path)

**n<sub>sims</sub>** = **100,000** ( $n_{\text{paths}}$  price paths per simulation)

**n<sub>steps</sub>** = **252 \* T** (1 monitoring point per day)

Fig. 6. Optimal numerical combination that minimizes sampling error.

#### E. Path-wise Greeks Derivations:

Equivalent implementations of the Monte Carlo experiment above were constructed in Python's Tensorflow framework. Tensorflow enables parallel computation for expensive tasks by sending packages to a Graphics Processing Unit (GPU) or a Tensor Processing Unit (TPU). Executing the algorithm within the Google Colaboratory Graphical User Interface allowed the use of Google's GPU and TPU units for the calculation of the numerical result. This enabled the computation of path-wise Greeks for both arithmetic and geometric average call and put options. Fig. 5 shows the exact python definition of the algorithm, as a pseudocode interpretation of the Tensorflow definitions was extremely verbose. The algorithm generates average values of Greeks across all simulations and stores them in arrays. Table I Illustrates the Greeks generated in a convenient mnemonic rule.

#### F. Black-Scholes-Merton Greeks

The definition has been updated to include numerical and graphical representations of Greeks in order to better assess the sensitivities of the options contracts under evaluation. Table I shows the BSM Greeks capabilities in the algorithm (underlined). In addition, it includes Rho ( $\rho$ ), or the sensitivity of an option or an options portfolio to changes in the risk-free rate interest rate, as well as Phi ( $\varphi$ ), or the expected change in the option premium due to small changes in the foreign currency interest rate, when applicable. All options calculations have been adjusted for the modified BSM model, and  $\Sigma_G$  (Eq. 17) in particular. Fig. 7 & 8 present the Greeks surfaces for a range of underlying price ( $S_0$ ) and maturity (T) values of a call and put options under the BSM model.

TABLE I. GREEKS CAPABILITIES OF THE MC & BSM ALGORITHMS.

	<i>Asset Price (S)</i>	<i>Volatility (<math>\sigma</math>)</i>	<i>Expiry (T)</i>
Option Price	<u>Delta (<math>\Delta</math>)</u>	<u>Vega (V)</u>	<u>Theta (<math>\Theta</math>)</u>
Delta ( $\Delta$ )	<u>Gamma (<math>\Gamma</math>)</u>	Vanna ( $D\Delta D\sigma$ )	<u>Charm (<math>D\Delta Dt</math>)</u>
Vega (V)	<u>Vanna (<math>DVDS</math>)</u>	<u>Volga (<math>DVD\sigma</math>)</u>	Veta ( $DVDt$ )
Gamma ( $\Gamma$ )	Speed ( $D\Gamma DS$ )	Zomma ( $D\Gamma D\sigma$ )	Color ( $D\Gamma Dt$ )
Vomma ( $DVD\sigma$ )	N/A	Ultima( $DVomD\sigma$ )	Totto ( $DVomDt$ )

### III. RESULTS & DISCUSSION

Asian options are rarely traded on an underlying stock of a publicly traded company within NYSE or NASDAQ making information on these options impossible to obtain from Yahoo Finance. However, these options are commonly used on commodities, most commonly on oil futures. This information is commonly available on CME (Chicago Mercantile Exchange). Due to the lack of a personal or corporate/institutional membership to their data service, using CME options data as a benchmark for the options proved to be difficult. However, within the scope of this research, the analytical results are computed and minimized through a series of cross-referencing and variance reduction techniques in order to estimate sufficiently accurate option prices without the need for data extraction with cumbersome data wrangling procedures. Fig. 17 shows the results of the lognormal MC engine. We see that as

#### A. Linetsky vs. Arithmetic MC

Linetsky [2] offers a set of test cases that has become the leading standard in the research literature with regards to calibration and accuracy test hypotheses. Using the eigenfunction expansion approach, the approximation yields results deemed to be the most accurate arithmetic average computation. The comparison of the Monte Carlo Arithmetic Average Options algorithm agrees with the test results within a range of absolute errors,

[0.0147%, 0.7345%]

14

The error was minimized first, by increasing the number of price paths generated by the GBM function; second, by incrementally increasing the number of Monte Carlo simulations; third, by determining the optimal number of timesteps given the available timeframe. The numerical combination to minimize the errors that stem from insufficient sampling are shown in Fig. 6 and were identified through trial-and-error. The combination was used in all experiments, unless shown otherwise.

For geometric average Asians we should get good agreement with the closed form result.

In the case of the Geometric Average pricing algorithm, the results are compared with the BSM model numerical outputs and graphs of Greeks and graphs of the call option values across time to expiry (T), incremental values of the risk-free interest rate, volatility cases and others. Fig. 10 illustrates the BSM model results and Fig. 12 illustrates the equivalent cases from the MC Call Options under the combination of MC model parameters in Fig. 6. The results for three cases of call and put options with strikes  $K = 95, 100, 105$  are summarized in Table II. In accordance with Ruttiens [3] and Vorst [4], employing the solution to the geometric Ref.[3] is Linetsky. (?)

average problem leads to underpricing in put and/or overpricing in call options. However, the errors are within the range,

$$[0.2047\%, 0.5438\%] \quad 15$$

which was deemed sufficient for the purposes of this benchmark.

TABLE II. GEOMETRIC MONTE CARLO PRICER & BSM BENCHMARKS.

<b>G<sub>type</sub>(K, T)</b>	<b>BSM</b>	<b>Monte Carlo (G<sub>n</sub>)</b>	<b>% error (ε)</b>
G <sub>c</sub> (95,1.0)	12.50853848101788	12.46260400000000	<b>0.367228</b>
G <sub>c</sub> (100,1.0)	9.61215966961383	9.638977665500253	<b>0.279001</b>
G <sub>c</sub> (105,1.0)	7.185865725921304	7.171154551742235	<b>0.204724</b>
G <sub>p</sub> (95,1.0)	2.194652455717919	2.198018800000000	<b>0.153388</b>
G <sub>p</sub> (100,1.0)	3.6018135264391495	3.610188200000000	<b>0.232514</b>
G <sub>p</sub> (105,1.0)	5.479059464871914	5.489564400000000	<b>0.191729</b>

### C. Variance Reduction

A variance reduction technique is employed to define the confidence intervals of the mean option price and to reduce the potential errors of averaging across each simulation and a The geometric average option payoff is used as a control variate to calibrate the accuracy of the arithmetic average component of the algorithm. This can be done by subtracting the geometric average option value from the equivalent arithmetically average counterpart, and then adding the geometrically averaged result back into the final option value. A second experiment was conducted using the difference between the option values of each averaging method and then tracking the confidence intervals at each step to obtain a descriptive set of the deviations between them. The differences become smaller along each timestep, further confirming the efficacy of the variance reduction technique in approximating the true mean of the option value. Using the same visualization script as the BSM results in III.B, the result of the control variates case of the MC method is evaluated and plotted against a range of market variables (Fig. 12).

### D. Other MC method error estimates

If the probability density of estimator  $y$  is  $f(y)$ , its expected value is,

$$\mu_y \equiv E(y) = \int_{-\infty}^{\infty} y f(y) dy \quad 16$$

For  $n_{sim}$  trials, we can use the sample mean as an estimator for  $\mu_y$ ,

$$\bar{y} \equiv \frac{1}{N} \sum_{i=1}^N y_i \quad 17$$

The moments of the error are,

$$\begin{aligned} &= E(\bar{y} - \mu_y) = E(\bar{y}) - \mu_y \\ &= E\left(\frac{1}{N} \sum_{i=1}^N y_i\right) - \mu_y \\ &= \frac{1}{N} \sum_{i=1}^N E(y_i) - \mu_y \end{aligned} \quad 19$$

Due to the random sampling aspects of Monte Carlo methods,  $E(\bar{y}) = \mu_y$ . Therefore,

$$E(\bar{y} - \mu_y) = \frac{1}{N} N \mu_y - \mu_y = 0 \quad 20$$

$$\mu_{\bar{y}} \equiv E(\bar{y}) \equiv \mu_y \quad 21$$

In other words, the error of evaluating  $\mu_y$  using  $\bar{y}$  as an estimator is zero, and the estimator  $\bar{y}$  is *unbiased*. For the variance,

$$\begin{aligned} Var(\bar{y} - \mu_y) &\Leftrightarrow \\ Var(\bar{y}) - Var(\mu_y) &\xrightarrow{Var(\mu_y)=0} \\ Var\left(\frac{\sum_{i=1}^N y_i}{N}\right) &\Leftrightarrow \\ \frac{1}{N^2} Var(\sum_{i=1}^N y_i) &\xrightarrow{Cov(y_i,y_j)=0} \\ \frac{1}{N^2} \sum_{i=1}^N Var(y_i) &\Leftrightarrow \\ \frac{1}{N^2} N \sigma_y^2 &\Leftrightarrow \\ \sigma_{\bar{y}}^2 = E[(\bar{y} - \mu_y)^2] &= \frac{\sigma_y^2}{N} \end{aligned} \quad 22$$

Eq. 22 illustrates the interesting relationship between the standard error  $\sigma_{\bar{y}}$  of the estimator and the sample size  $N$  i.e. the standard error decreases with  $\sqrt{N}$ . In terms of the Monte Carlo Engine, this finding aids in reducing unnecessary computation and setting bounds on the minimum number of paths needed to achieve optimality. To reduce the standard error in the experiment, the sample size was increased by a factor of 1,000 from the original sample. For the variance reduction implementation, the sample size was increased by a factor of 100 for satisfactory results, and to meet computational constraints.

this analysis should follow the same option on 5 consecutive days

## E. Hedging Analysis

The analysis involves a portfolio of 5 Asian call options with stock underlying prices  $S_i$  generated by the GBM price path generator. The following assumptions are made,

- Each path is refined to approximately 11 updates/second. The calculation assumes that price movements occur evenly throughout a 24-hour cycle.
- After-hours trading does not move the closing price of the security each day i.e. the price of the security at 9:30am (open market) is the same as the closing price the previous day.
- The security pays no dividend.

Each option represents a contract for one of 5 business days. Using the Greek letter derivatives property of the pricing algorithm, the static call option delta,  $\Delta(t_i)$  is obtained for each day and is used to create a second portfolio with option values,

$$C(t_i) - \Delta(t_i)S_i \quad 23$$

The portfolio construction based on Eq. 23 is called a *delta-hedged portfolio* and exhibits the unique property of protecting the bundle against directional risk from volatility changes in the price of the underlying asset. The Asian call option delta is shown in Figure 3; it represents the rate of change of the option premium when the underlying asset price changes, and the number of shares required to maintain the overall traders' position delta neutral.

The  $C(t_i)$  plot of the two portfolios across time illustrates the additional protection of a delta-hedged portfolio (Fig. 13 and 14) as seen in the greatly reduced variability of the hedged portfolio throughout the week, as opposed to the naked portfolio. Asian options are easier to hedge than regular options as the payoff from an Asian option becomes more certain with the passage of time. As a result, the amount of uncertainty that needs to be hedged decreases with the passage of time. Delta-hedging is a very common options strategy and is very often paired with gamma hedging in a delta-gamma hedging strategy, which combines both delta and gamma hedges to protect against the risk of price changes in the underlying asset and the delta itself.

## IV. NEXT STEPS

Time permitting the team would have liked to incorporate additional properties to the algorithm in order to more accurately reflect on the price dynamics of an Asian stock. One potential path forward is the addition of a volatility surface feature, which would generate more realistic results than those of a Lévy-approximation

**The BS model does not have a volatility surface, as the vol is constant. Models like Local volatility and Stochastic vol models allow a vol surface.**

based model (where the volatility is static across the entire surface i.e. the plane,  $z = \sigma\sqrt{3}$ ). Finally, we soon hope to test the pricing algorithm against real data and to take advantage of the unprecedented circumstances in the oil futures markets by incorporating data and features relevant to valuation in similar black-swan events.

Other Asian Options pricing algorithms developed over the years include, but are by no means limited to:

- Path Integral Approach - Effective Classical Potential (VG is an alternative asset price model based on jumps while BS is a diffusion model)
- Rogers & Shi's PDE
- Variance Gamma Model
- Lévy Moment Matching Method

It would be a fascinating endeavor to incorporate more functionality in the pricing algorithm and be able to price options more accurately and efficiently. The hope of this paper is to generate additional questions and research on the topic and provide more efficient tools and methodologies to industry professionals.

**G\_c may be left over from a previous version of the table.**

TABLE III. ARITHMETIC MONTE CARLO PRICER & LINETSKY BENCHMARKS.

Case	r	$\sigma$	$S_0$	Linetsky	Monte Carlo	% ε
$G_c(K,T)_1$	0.0200	0.10	2.0	0.0559860415	0.0559778385	0.0147
$G_c(K,T)_2$	0.1800	0.30	2.0	0.2183875466	0.2170470505	0.6138
$G_c(K,T)_3$	0.0125	0.25	2.0	0.1722687410	0.1726129894	0.1998
$G_c(K,T)_4$	0.0500	0.50	1.9	0.1931737903	0.1917548655	0.7345
$G_c(K,T)_5$	0.0500	0.50	2.0	0.2464156905	0.2472542514	0.3403
$G_c(K,T)_6$	0.0500	0.50	2.1	0.3062203648	0.3066361640	0.1357
$G_c(K,T)_7$	0.0500	0.50	2.0	0.3500952190	0.3475691276	0.7215

**very good agreement!**

## ACKNOWLEDGMENTS

We would like to thank Professor Dan Pirjol for his invaluable guidance throughout this project; the Stevens Financial Engineering Department for providing us with remote Bloomberg Terminal access; and the Google Colaboratory and Google Compute service providers for enabling large-scale options pricing models at minimal costs for researchers.

## AUTHORS

**Theo Dimitrasopoulos**– MSc. Financial Engineering, Stevens Institute of Technology 2021; Bachelor of Science in Engineering, Structural Engineering, Princeton University 2017; [theonovak.com](http://theonovak.com); [tdimitr1@stevens.edu](mailto:tdimitr1@stevens.edu)

**William Kraemer**– MSc. Financial Engineering, Stevens  
Institute of Technology 2020; BEng. Engineering Management,  
Stevens Institute of Technology 2019; *wkraemer@stevens.edu*

**Snehal Rajguru**– MEng. Engineering Management, Stevens  
Institute of Technology 2021; BEng. Electronics and  
Telecommunication Engineering, Pune University 2018;  
*srajguru@stevens.edu*

**Vaikunth Sheshadri**– Master of Science in Financial  
Engineering, Stevens Institute of Technology 2021; Bachelor of  
Science in Industrial Engineering, University of Wisconsin-  
Madison 2019; *vshesadr@stevens.edu*

## REFERENCES

- [1] E. Lévy, 1992 "Pricing European average rate currency options". Journal of International Money and Finance, 1992, vol. 11 pp. 474-491.
- [2] Kemna, A.G.Z. & Vorst, A.C.F. 1990, "A pricing method for options based on average asset values". Journal of Banking & Finance, Elsevier, March 1990, vol. 14(1), pp 113-129.
- [3] V. Linetsky (2002), Spectral Decomposition of the Option Value. Published as "[Exotic Spectra](#)", RISK, April 2002, pp.85-89.
- [4] Laura Ballotta, Russell Gerrard & Ioannis Kyriakou (2017) Hedging of Asian options under exponential Lévy models: computation and performance, The European Journal of Finance, 23:4, 297-323.
- [5] Yor, M. (2001). Exponential Functionals of Brownian Motion and Related Processes. Springer-Verlag, New York.

## APPENDIX:

→

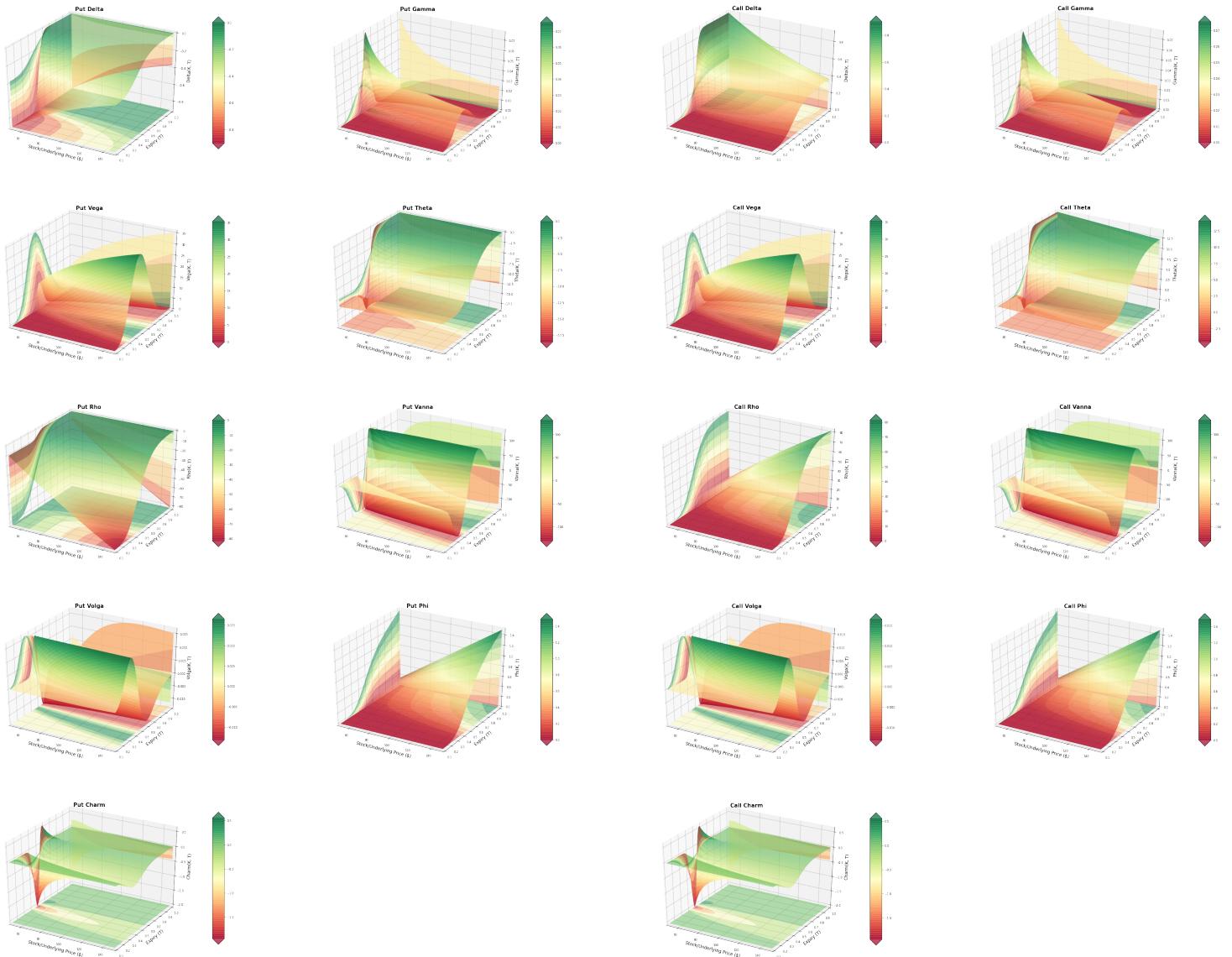


Fig. 8. Graphs of relevant BSM Greeks for a range of underlying price ( $S_0$ ) and maturity (T) values of Asian Put options.

Fig. 7. Graphs of relevant BSM Greeks for a range of underlying price ( $S_0$ ) and maturity (T) values of Asian Call options.

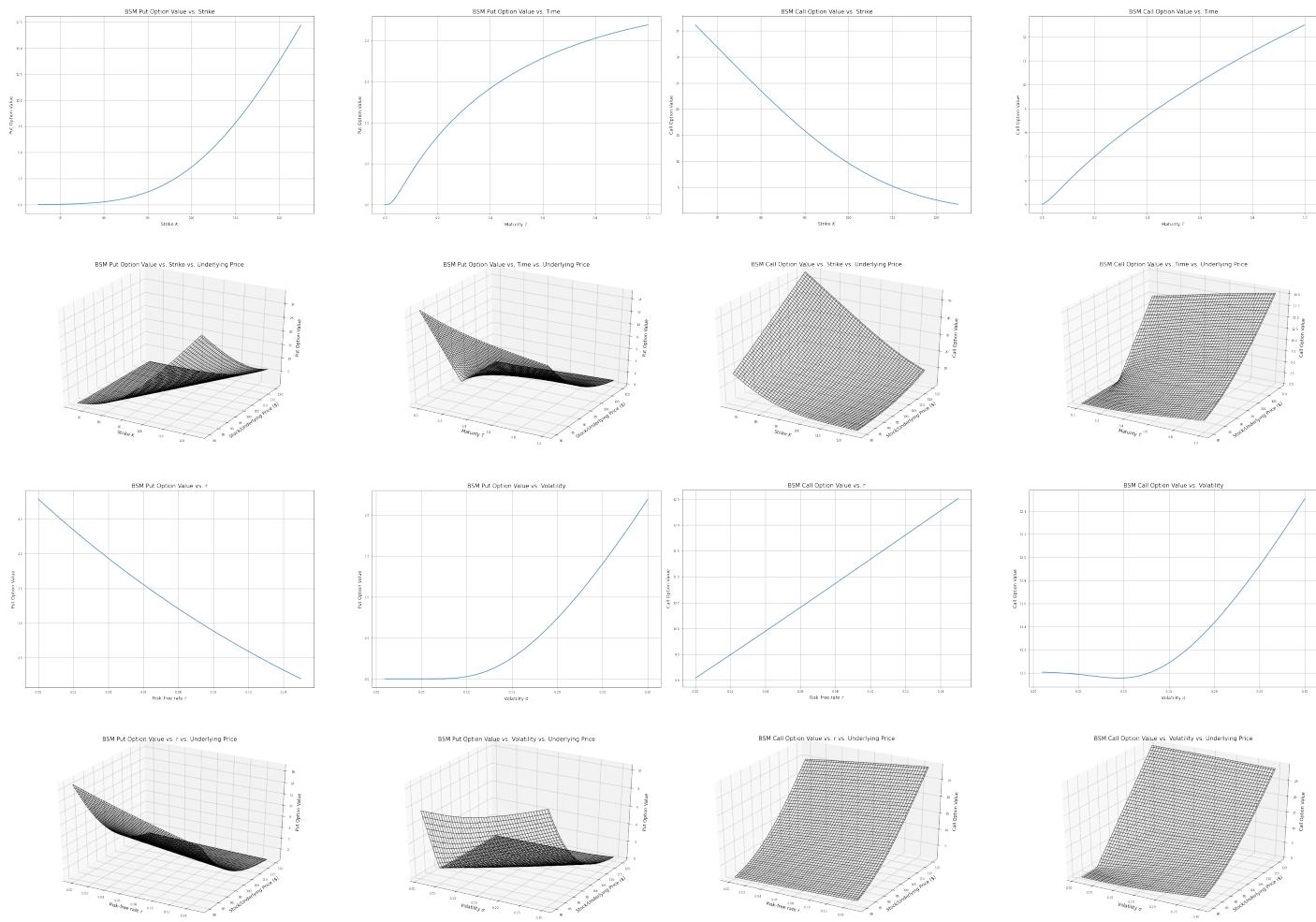


Fig. 9. Graphs of relevant BSM Greeks for a range of underlying price ( $S_0$ ) and maturity ( $T$ ) values of Asian Call & Put options.

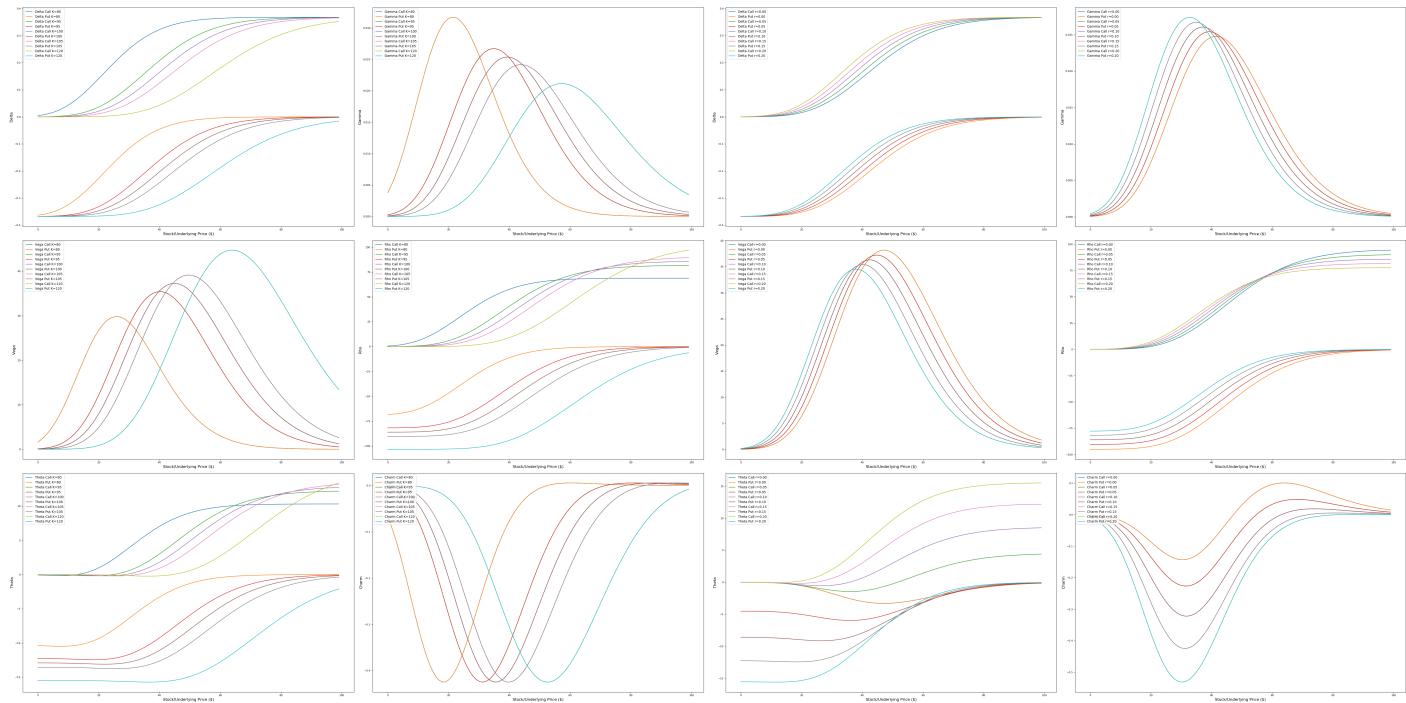


Fig. 10. Graphs of relevant BSM Greeks for three strikes  $K=95, 100, 105$  and risk-free rates for Asian Call and put options.

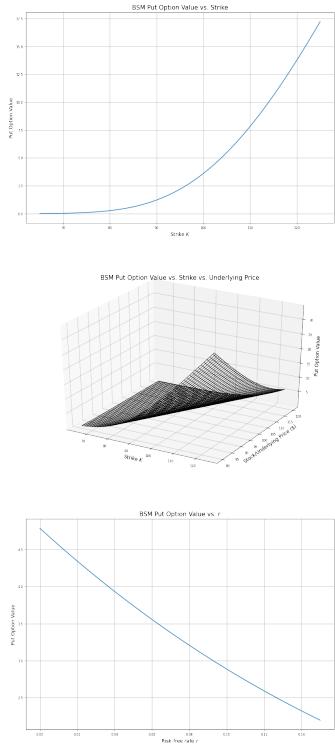


Fig. 11. Graphs of relevant BSM Greeks for a range of underlying price ( $S_0$ ) and maturity ( $T$ ) values of Asian Call options using the Geometric Average Monte Carlo simulation.

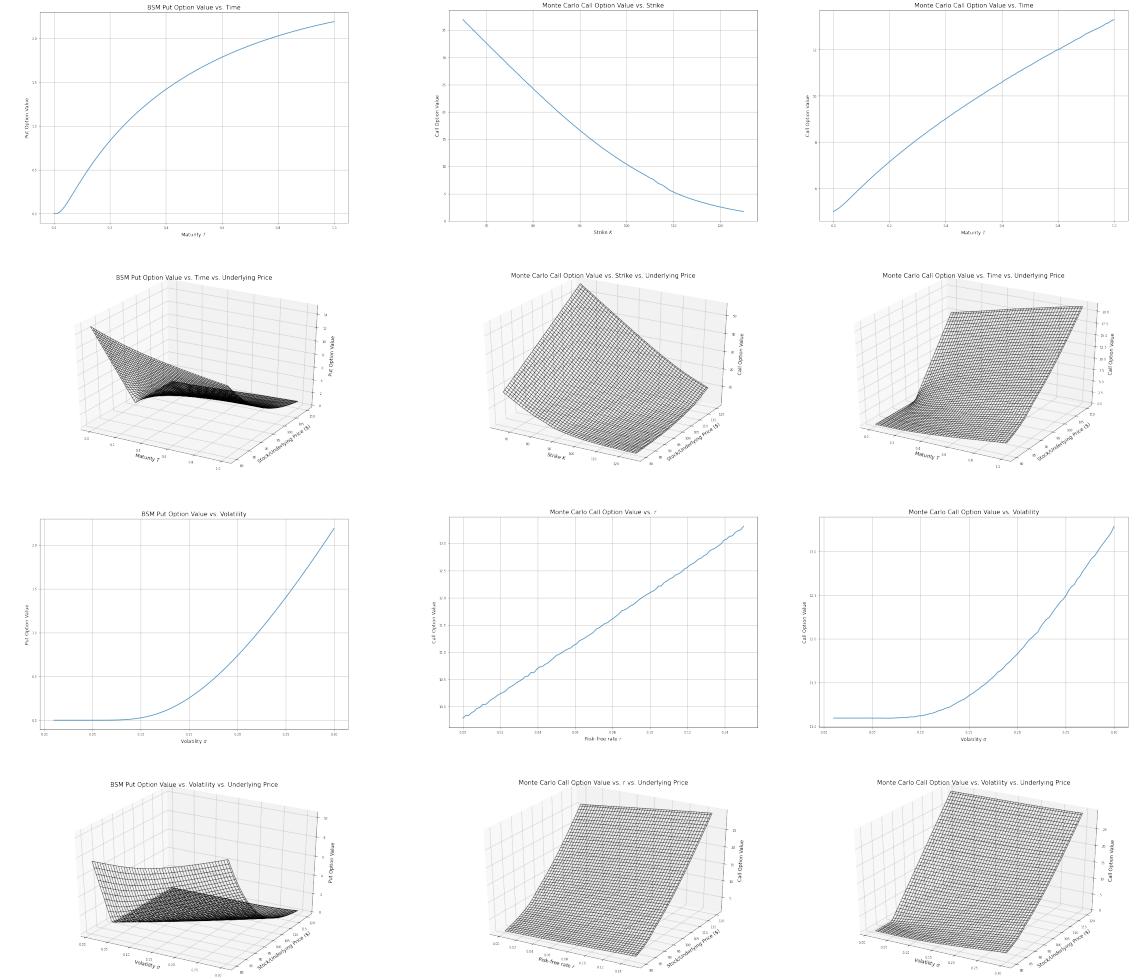


Fig. 12. Graphs of relevant BSM Greeks for a range of underlying price ( $S_0$ ) and maturity ( $T$ ) values of Asian Call options using the Geometric Average Monte Carlo simulation with the control variates reduction technique.

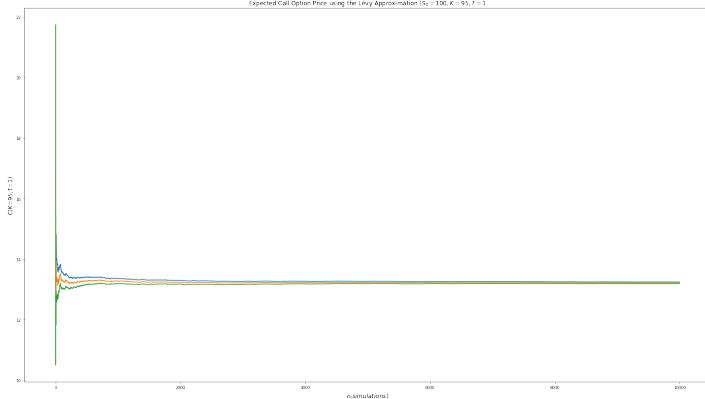


Fig. 17. Call option price utilizing the Lévy Approximation in the MC Lognormal Engine. The x-axis shows the  $n_{\text{simulations}}$  and the y-axis shows the  $E[C(K = 95, T = 1)]$ . It is worth noting the fast convergence of the expected price.

The convergence is to what theoretical price? Should be to the Levy approximation result.

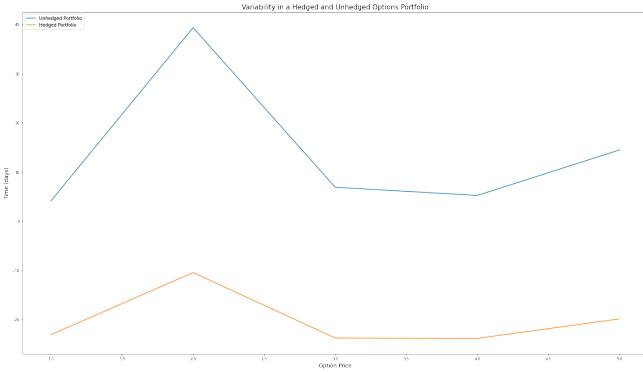


Fig. 13. Call Options Prices across 5 days. Each point ( $S_0$ ) was an input to a secondary MC simulation that generates the next point (hedged portfolio shifted for readability; actual starting point is the starting point of the naked portfolio (top)

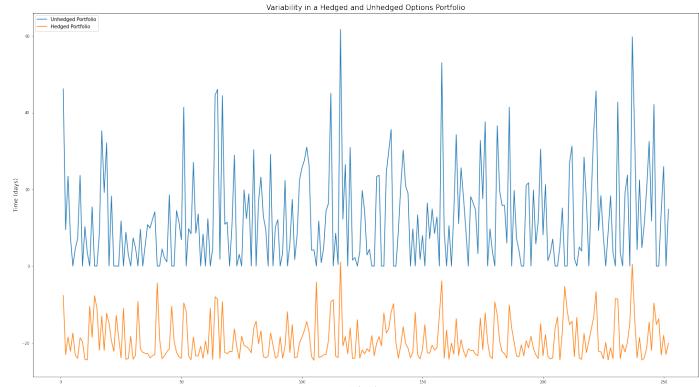


Fig. 14. Call options prices to expiration date. (similar to Fig 14, the hedged portfolio was shifted down for readability)



Fig. 16. CL1 Crude Oil Futures price fell in negative territories for the first time in history. April 20<sup>th</sup>, 2020. Bloomberg Terminal.

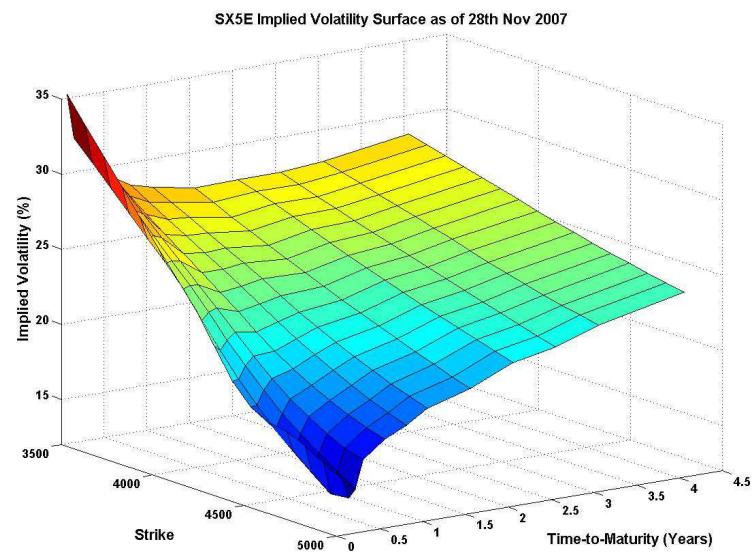


Fig. 15. The EURO STOXX 50 volatility surface. Source: IEOR 4707, Financial Engineering, Continuous-Time Models. Martin Haugh, Columbia University, 2013.

## DERIVATIONS

The following is the calculation of the first and second moments i.e. the mean and variance of the general Martingale shown in Eq. 11:

$$S_{t+\Delta t} = S_t * e^{\sigma \varepsilon - \frac{\sigma^2}{2} t}$$

The mean ( $M_1$ ) of  $\int_0^T S_t dt$  in  $\mathcal{P}^*$ :

$$\begin{aligned} \mathbb{E}^* \left[ \frac{1}{T} \int_0^T S_t dt \right] &= \mathbb{E}^* \left[ \frac{1}{T} \int_0^T e^{rt} S_0 \exp(\sigma W_t - \sigma^2 t/2) dt \right] \\ &= \mathbb{E}^* \left[ \frac{S_0}{T} \int_0^T e^{rt} M_t dt \right] \\ &= \frac{S_0}{T} \mathbb{E}^* \left[ \int_0^T e^{rt} M_t dt \right] \\ &= \frac{S_0}{T} \int_0^T e^{rt} \mathbb{E}^* [M_t] dt \\ &= \frac{(e^{rT} - 1)}{rT} S_0 \quad (M_t = \exp(\sigma W_t - \sigma^2 t/2)) \end{aligned}$$

>>Imported from MacTeX

The variance ( $M_2$ ) of  $\int_0^T S_t dt$  in  $\mathcal{P}^*$ :

$$\begin{aligned} \text{Var} \left[ \frac{1}{T} \int_0^T S_t dt \right] &= \mathbb{E}^* \left[ \left( \frac{1}{T} \int_0^T S_t dt \right)^2 \right] - \left( \mathbb{E}^* \left[ \frac{1}{T} \int_0^T S_t dt \right] \right)^2 \\ &= \mathbb{E}^* \left[ \left( \frac{1}{T} \int_0^T S_t dt \right)^2 \right] - \left( \frac{(e^{rT} - 1)}{rT} S_0 \right)^2 \\ \mathbb{E}^* \left[ \left( \frac{1}{T} \int_0^T S_t dt \right)^2 \right] &= \mathbb{E}^* \left[ \left( \frac{1}{T} \int_0^T S_u du \right) \left( \frac{1}{T} \int_0^T S_v dv \right) \right] \\ &= \mathbb{E}^* \left[ \left( \frac{1}{T^2} \int_0^T \int_0^T S_u S_v du dv \right) \right] \\ &= \frac{1}{T^2} \int_0^T \int_0^T \mathbb{E}^*[S_u S_v] du dv \quad (\text{Fubini}) \\ &= \frac{1}{T^2} \int_0^T \int_0^T \mathbb{E}^* \left[ \frac{S_u}{S_v} S_v^2 \right] du dv \\ &= \frac{2}{T^2} \int_0^T \int_0^u \mathbb{E}^* \left[ \frac{S_u}{S_v} \right] \mathbb{E}^* [S_v^2] dv du \quad (\text{Independent variables}) \\ &= \frac{2}{T^2} \int_0^T \int_0^u \mathbb{E}^* \left[ e^{r(u-v)} e^{\sigma(W_u - W_v) - \frac{\sigma^2}{2}(u-v)} \right] \mathbb{E}^* \left[ S_0^2 e^{2rv} e^{2\sigma W_v - \frac{2\sigma^2}{2}v} \right] dv du \\ &= \frac{2}{T^2} \int_0^T \int_0^u e^{r(u-v)} \mathbb{E}^* \left[ e^{\sigma(W_u - W_v) - \frac{\sigma^2}{2}(u-v)} \right] S_0^2 e^{2rv} e^{\sigma^2 v} \mathbb{E}^* \left[ e^{2\sigma W_v - \frac{(2\sigma)^2}{2}v} \right] dv du \\ &= \frac{2S_0^2}{T^2} \int_0^T \int_0^u e^{r(u-v)} e^{2rv} e^{\sigma^2 v} dv du = \frac{2S_0^2}{T^2} \int_0^T \int_0^u e^{ru} e^{(r+\sigma^2)v} dv du \\ &= \frac{2S_0^2}{T^2} \int_0^T e^{ru} \frac{e^{(r+\sigma^2)u} - 1}{r + \sigma^2} du = \frac{2S_0^2}{T^2(r + \sigma^2)} \int_0^T e^{(2r + \sigma^2)u} - e^{ru} du \\ &= \frac{2S_0^2}{T^2(r + \sigma^2)} \left[ \frac{e^{(2r + \sigma^2)T} - 1}{2r + \sigma^2} - \frac{e^{rT} - 1}{r} \right] \end{aligned}$$

$$\text{Var} \left[ \frac{1}{T} \int_0^T S_t dt \right] = \frac{2S_0^2}{T^2(r + \sigma^2)} \left[ \frac{e^{(2r + \sigma^2)T} - 1}{2r + \sigma^2} - \frac{e^{rT} - 1}{r} \right] - \left( \frac{(e^{rT} - 1)}{rT} \right)^2 \quad \textcolor{red}{-1)^{\wedge} 2}$$

>>Imported from MacTeX