Comparing Stochastic Volatility Option Pricing Models

Cian Besler-Chalcroft, 30073046
Ryley Bosman, 30036791
Kendra O'Hearn, 30116424
Yiqi Zhao, 30072843

April 11, 2025

Abstract

Given the uncertainty and volatility of financial markets, it can be challenging to accurately price financial derivatives, a problem that is fundamental to financial mathematics. Although widely popular and still used today, the Black-Scholes model has a major drawback: constant volatility. In this project, we investigate the development of the stochastic volatility models designed to remedy this problem. Specifically, we will present the Heston model and two modified versions: Heston with Jumps and Rough Heston. To compare the performance of these models, we analyzed their implied volatility relative to real market behaviour and researched the accuracy of their option pricing results. Through our own Black-Scholes model implementation in Python, we priced put and call options based on market data and demonstrated the resulting implied volatility smile. Further, we briefly discuss calibration methods and the challenge it poses for increasingly complex models. We conclude by discussing the best use scenarios of these models and how different qualities of the models should be considered.

1 Introduction

The market for financial derivatives plays a significant role in the global economy and has seen incredible growth since the 1970's [1]. Due to their critical role in the economy, the correct pricing of derivatives is crucial. Otherwise, the market can become inefficient, risky, and unstable, which has significant economic consequences. The fear that stocks and option contracts were mispriced is one cause of the devastating 1987 stock market crash, known as 'Black Monday'. The impact this had on the economy emphasizes the need to accurately price derivatives. In this project, we investigate how a class of derivatives, European options, are priced and modeled. We take particular interest in volatility considerations of these models and how they both reflect reality and influence pricing accuracy.

An option is a financial instrument based on an underlying risky asset that provides investors opportunities to optimize their portfolios. Further, options are contracts that provide a chance to profit, and so there must be a price to enter this contract. Since the value of an option is unknown at a future time due to uncertainty of the underlying asset, determining an appropriate price of such a contract can be challenging. Solving this prob-

lem is one of the most important topics in financial mathematics. The Black-Scholes model for option pricing introduced in 1973 was transformative to this field and still holds great prevalence today. However, 'Black Monday' exposed the assumption of constant volatility in the Black-Scholes model as being inaccurate. This inaccuracy is obvious when the volatility implied by true market prices under the model is not constant. We implemented the Black-Scholes model in Python and used numerical methods to compute the implied volatility. The resulting volatility smile shows the constant volatility assumption is inaccurate.
One way to improve the volatility definition is to instead consider a stochastic process, that is, a variable that is random over time. In 1993, a closed form solution for option pricing that assumed stochastic volatility was published by Steven Heston, however, one of the downfalls of the Heston model is its poor performance over short time horizons. We investigate two modifications of the Heston model that aim to improve the accuracy of short-term pricing. The costs of improved accuracy in a stochastic volatility model often include a more demanding calibration process of the model parameters and an overall increase in computational complexity. This means that it is more challenging to use and understand these models when compared to Black-Scholes. Further, the popularity of Black-Scholes means information and implementations of this model are more widely available and easy to use. In this project, we compare how these models perform for option pricing over different time horizons and moneyness, which is a relationship between the price of the underlying asset and the strike price. Further, we take interest in how their volatility behaviour and resulting implied volatility compare to the true market volatility. Finally, we aim to conclude in which scenarios each model is most appropriate to use.

2 Background

We will first set the mathematical foundations and definitions that will be used throughout this project. This includes important underlying market assumptions, the definition of European put and call options, and an introduction to Brownian Motion and some fundamental concepts of stochastic processes and calculus.

2.1 Market Assumptions

All market models require some assumptions about the market. In this section, we will describe these assumptions and define a class of financial derivatives, European options, that provide the foundation of the models we investigate in this project.

For this project, we limit our scope to consider a market with only one risky asset or stock and one risk-free asset, with constant and continuously compounding interest rate r.

We assume the following unless otherwise stated:

- There are no transaction costs
- Stock pays no dividends
- The market is arbitrage free by the No Arbitrage Principle
- An investor may hold any real-numbered quantity of an asset, allowing short-selling

The No Arbitrage Principle (NAP) states that there does not exist an opportunity for risk-free profit with 0 initial investment in the market. This is an important assumption which is necessary for pricing models to be defined mathematically. Although the market is not truly arbitrage free, these opportunities are few and short-lived, making this assumption reasonable and not a source of error.

A European option is a financial derivative that specifies a stock S, strike price K, and expiry time T. We define them and their payoffs as the following:

Definition. A European call option gives its holder the right, but no obligation, to buy a share of stock S for a price K at expiry time T. It has payoff function, at expiry T, defined by

$$C(T, S_T) = max(S_T - K, 0)$$

Definition. A European put option gives its holder the right, but no obligation, to sell a share of stock S for a price K at expiry time T. It has payoff function, at expiry T, defined by

$$P(T, S_T) = max(K - S_T, 0)$$

An important relationship between the values of a call and put option at some time t, $0 \leq t \leq T$, is known as put-call parity given by:

$$C(t, S_t) + Ke^{-r(T-t)} = P(t, S_t) + S_t$$

A simple argument for this relationship is rooted in the NAP, as demonstrated by Desmond Higham [2].

Considering the above definitions of European options, one can observe that their payoff is at least 0 with a chance to be profitable. Then, by the NAP, this asset must have some cost to enter the contract. This conclusion leads to a fundamental problem: what should this option premium be? In Section 5, we compare different models used to determine these prices.

2.2 Brownian Motion and Stochastic Processes

Brownian Motion is used in many disciplines to describe random movements of objects. First used in financial mathematics by Louis Bachelier, Brownian Motion is used to model the random behaviour of stock prices.

Definition. A Standard Brownian Motion, denoted $(W_t)_{0 \leq t}$ is a continuous, but not differentiable, random variable on probability space $(\Omega, \mathcal{F}, P)$ defined by the following properties:

- $W_0 = 0$
- for $0 \leq s < t < \infty$, $W_t - W_s$ is normally distributed with mean 0 and variance t-s
- for all n such that $0 \leq t_{n-1} < t_n < t_{n+1} < \infty$, the increments $W_{t_{n+1}} - W_{t_n}$ and $W_{t_n} - W_{t_{n-1}}$ are independent

Capiński et al. justify this definition using the Central Limit Theorem [3]. Brownian Motion is used as the foundation of stochastic processes, representing their randomness.

Definition. A stochastic or Ito process $(X_t)_{0 \leq t}$ is a random variable on probability space $(\Omega, \mathcal{F}, P)$ defined by

$$X_t = X_0 + \int_0^t a(s)ds + \int_0^t b(s)dW_s \quad \text{for } t \in [0, T]$$

which is commonly written in the notation:

$$dX_t = a(t)dt + b(t)dW_t \qquad [3]$$

It is important to note that Brownian Motion, $(W_t)_{0\leq t}$, is not differentiable and the above is a different representation of the integral. Various Ito processes are used to represent stock prices and volatility, as we explore more in Sections 3 and 4.

From this definition, we introduce one of the most important theorems in stochastic calculus, the Ito Formula, also known as Ito's Lemma. This result is reached by using the chain rule and adjusting for quadratic variation. This adjustment appears in the final integral term in the theorem below [9].

Theorem (Ito Formula)**.** *Let $(X_t)_{0\leq t}$ be a stochastic process and $f(t, X_t)$ be a sufficiently differentiable function. Then,*

$$f(t, X_t) - f(0, X_0) = \int_0^t \partial s f(s, X_s)ds + \int_0^t \partial x f(s, X_s)dX_s + \frac{1}{2}\int_0^t \partial_x^2 f(s, X_s)b(s)^2 ds$$

This theorem is used to compute and justify various stochastic processes and differential equations that we investigate in Section 4.

Another important concept in derivative pricing is the Risk-Neutral Probability Measure, denoted $\tilde{P}$. Under this probability measure, the expected value of the discounted future asset price is equal to the current asset price. That is,

$$S_t = E_{\tilde{P}}[e^{-r(T-t)}S_T]$$

where $e^{-r(T-t)}$ is the discount factor. This measure is important for asset pricing theorems and its existence implies that the NAP holds. A detailed explanation and justification of the Risk-Neutral Measure is given by Iyer [9].

3 Black-Scholes Model

In 1973 Fisher Black and Myron Scholes published a model to compute the price of European put and call options, which became foundational to financial mathematics [4]. Like all models, it has drawbacks, which has motivated the investigation and development of new models, which we begin to explore in Section 4.

The Black-Scholes (B-S) model requires the market assumptions outlined in Section 2.1 and has the following parameters:

- Stock price $(S_t)_{0\leq t}$, simplified as S, which follows a Geometric Brownian Motion
- Constant interest rate r
- Strike price K
- Time to expiry T-t
- Volatility σ, defined as the standard deviation of the log stock return

Representing a stock as a Geometric Brownian Motion satisfying the following definition results in log-normal returns with constant volatility σ.

Definition. A Geometric Brownian Motion (GBM), a stochastic process denoted $(S_t)_{0\leq t}$, where S_t is the value of the process at time t and is defined by

$$S_t = S_0 exp[(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t]$$

satisfying the stochastic differential equation (SDE):

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where μ, σ are the mean and standard deviation of the log return, respectively and W_t is a standard Brownian Motion.

Black and Scholes derived the following PDE with solution $V(t, S)$, an option valuation formula:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

with boundary conditions [6]:

- $V(0, S) = 0$,
- $V(T, S) =$ payoff(S), the payoff function of the option
- $V(t, S) \to e^{-r(T-t)}V(T, 0)$ as $S \to 0^+$
- $S^2 \frac{\partial^2 V}{\partial S^2} \to 0$ as $S \to \infty$

on the domain: $[0, T]$x$[0, \infty)$

The derivation and solution of this differential equation is beyond the scope of this project, however, we state the resulting formulas for the value of a European call and put option [4]:

$$C(t, S_t) = N(d_+)S_t - N(d_-)Ke^{-r(T-t)}$$
$$P(t, S_t) = N(-d_-)Ke^{-r(T-t)} - N(-d_+)S_t$$

where $N(x)$ is the cumulative normal distribution $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-z^2/2}\,dz$ and

$$d_+ = \frac{1}{\sigma\sqrt{T-t}}[ln(\frac{S_t}{K}) + (r + \frac{1}{2}\sigma^2)(T-t))]$$

$$d_- = \frac{1}{\sigma\sqrt{T-t}}[ln(\frac{S_t}{K}) + (r - \frac{1}{2}\sigma^2)(T-t)]$$

In the Black-Scholes model, volatility, σ, is assumed constant, which is of particular interest in this project. Further, volatility cannot be directly observed in this model nor is there a closed-form solution, meaning it may only be computed implicitly. This implied volatility is the volatility which the market implies under a given model, it is not the volatility which the model defines.

For a call option, one way to compute the implied volatility is by rewriting the B-S call option formula, under the risk-neutral measure, as a function of σ, resulting in:

$$f(\sigma) = S_0 N(d_+(\sigma)) - Ke^{-r(T-t)}N(d_-(\sigma)) - C$$

where C is the observed call option price.

Implied volatility, $\hat{\sigma}$, is defined as the root of $f(\hat{\sigma}) = 0$ and may be computed using various numerical root finding techniques, including the bisection method and the Newton-Raphson method [6]. A similar formula for a put option exists and can be found using the principle of put-call parity. We explore this in further detail in Section 5.4 and implemented this process using real market data to demonstrate the Black-Scholes implied volatility smile in Figure 9.

The B-S model's assumption of constant volatility is kept over different strike prices and time to expiry. However, this contradicts the observed behaviour in the market, as demonstrated in Figure 1. These trends are known as volatility smiles or surfaces, however another popular measure is called the volatility skew, which is the spread or difference in volatility between options with different strikes. In practice, different relationships with implied volatility help traders make informed hedging decisions to mitigate risk [5].

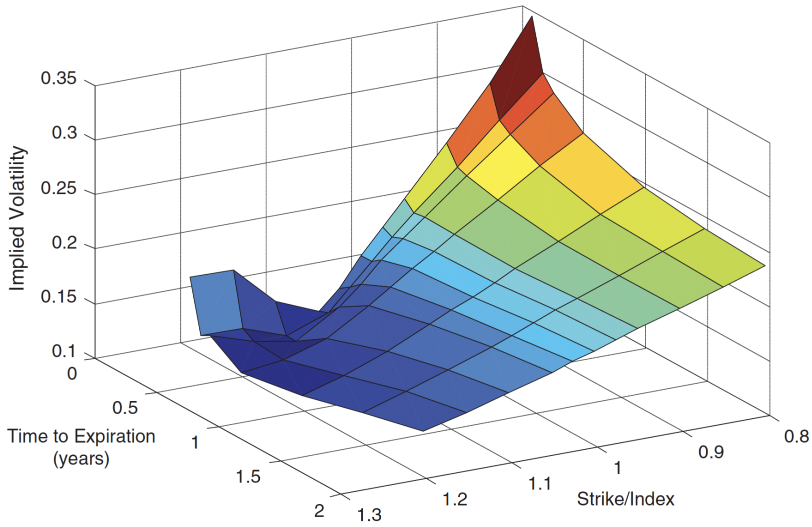

Figure 1: Implied Volatility Surface, S&P 500, December 31, 2015 [5]

The above surface was generated from S&P 500 options from 2015 and clearly demonstrates that the constant volatility assumption does not hold in real markets.

Other observed market behaviours, like the Leverage Effect, which describes an inverse relationship between stock price and volatility, and the Clustering Effect, which means that large changes in the market are followed by large changes and small changes are followed by small changes, are not considered or demonstrated by the B-S model [10].

These drawbacks due to the constant volatility assumption, which we investigate further in Section 5.1, led to the development of different models that use stochastic volatility in their representation of stock prices and to value options, which we demonstrate in Section 4.

4 Stochastic Volatility Models

In the years since Black and Scholes published their model, its disadvantages have been examined, and new models have been developed to address these issues. The most popular adjustment is to abandon the definition of the stock price process as a Geometric Brownian Motion with constant volatility. Instead, the stock price process is adjusted to allow volatility to be defined as a stochastic process in an attempt to make the model more realistic. Although this idea of stochastic volatility existed earlier, with related papers written shortly after B-S was published, Steven Heston was the first to propose a closed form solution of a stochastic volatility model in 1993 [8]. Two popular modifications to the Heston model exist to remedy it's poor short-term performance. The first involves adding log-normal random 'jumps' to the stock price process, and the other modifies the volatility process by making it 'rough'. In this section, we will define these three stochastic volatility models. Then, in Section 5, we will compare their performance, with a focus on moneyness and time-horizons.

4.1 The Heston Model

Following Steven Heston's 1993 publication, the Heston model has become arguably the most important model for pricing options. Its importance can be traced back to the stock market crash of October 1987, commonly referred to as Black Monday. Following the crash, the smiles and skews of the implied volatility surface were significantly amplified, thus bringing into question the Black-Scholes model's ability to produce sufficiently accurate option prices. Additionally, it highlighted the restrictive nature of the Black-Scholes model's market assumptions. The most fragile of these is the assumption that continuously compounded stock returns exhibit a normal distribution with constant volatility. Several empirical studies since 1987 have shown that this assumption does not hold for equity markets. It is now evident that returns are not normally distributed. Instead, they exhibit skewness and a fat-tailed kurtosis, which normality does not account for. Volatility is not constant but is often inversely correlated to price, with higher prices tending to exhibit lower volatility than low stock prices. Heston allows for time-varying volatility in his model by defining volatility as its own stochastic process. The models of Hull and White (1987), Scott (1987), and Wiggins (1987) are among the most notable Stochastic Volatility models to predate Heston. The Heston model was not the first Stochastic Volatility model. However, it has become the most important for pricing options and is a benchmark against which new Stochastic Volatility models are compared [7].

The Heston model proposes a definition of the stock price process $(S_t)_{0 \leq t}$ that satisfies the PDE below, which is similar to the GBM used in B-S model, but volatility is defined as a stochastic process, $(\nu_t)_{0 \leq t}$.

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t W_t^S$$

where ν_t satisfies $d\sqrt{\nu_t} = -\beta\sqrt{\nu_t}dt + \delta dW_t^\nu$

It can be shown by using Ito's Lemma [7] that $d\nu_t = \lambda(\theta - \nu_t)dt + \xi\sqrt{\nu_t}dW_t^\nu$

It is often convenient to use Ito's lemma again defining $X_t = ln(S_t)$ to obtain the log price process:

$$dX_t = \left(u - \frac{1}{2}\right) dt + \sqrt{\nu_t} dW_t^\nu$$

The parameters for the stock price model can be interpreted as [6]:

- μ: the drift term of the stock process S_t, a constant real number;

- $\lambda = 2\beta$: the mean reversion speed for the volatility process, a positive constant;
- $\theta = \frac{\delta^2}{2\beta}$: the mean reversion level of the volatility process, a positive constant;
- $\xi = 2\delta$: the volatility of the volatility process or vol of vol;
- ν_0: the initial term of the volatility process;
- ρ: the correlation coefficient between the two Brownian motions W_t^S and W_t^ν, a real number between $[-1, 1]$.

The European call Price in the Heston model can be formulated as a Black-Scholes like equation [8]:

$$C(t, S_t) = S_t P_1 - K e^{-r(T-t)} P_2$$

where P_1 and P_2 are the conditional probabilities that the option expires in the money. An option expires 'in the money' if its resulting payoff is positive. That is $S_T > K$ for a call or $S_T < K$ for a put. Although these probabilities are not 'immediately available in closed-form' [8], P_1 and P_2 must satisfy the following partial differential equation:

$$\frac{\partial P_j}{\partial_t} + \rho\sigma\nu\frac{\partial^2 P_j}{\partial\nu\partial_x} + \frac{1}{2}\nu\frac{\partial^2 P_j}{\partial x^2} + \frac{1}{2}\sigma^2\nu\frac{\partial^2 P_j}{\partial\nu^2} + (r + u_j\nu)\frac{\partial P_j}{\partial x} + (a - b_j\nu)\frac{\partial P_j}{\partial\nu} = 0$$

For j =1,2 where $u_1 = \frac{1}{2}, u_2 = \frac{-1}{2}, a = \kappa\theta, b_1 = \kappa + \lambda - \rho\sigma$, and $b_2 = \kappa + \lambda$.

An additional reason why the Heston model is so revolutionary is the fact that it was the first model to price options using the characteristic function.

Heston (1993) [8] showed the characteristic function solution for the log stock price of the Heston model to be:

$$f_j(X, \nu, t; \phi) = exp(C_j(T - t; \phi) + D_j(T - t; \phi)_j\nu + i\phi x)$$

where $i = \sqrt{-1}$ is the imaginary unit

$$C(\tau; \phi) = r\phi i\tau + \frac{a}{\sigma^2}\left[(b_j - \rho\sigma\phi i + d)\tau - 2ln\left[\frac{1 - ge^{d\tau}}{1 - g}\right]\right],$$

$$D(\tau; \phi) = \frac{b_j - \rho\sigma\phi i + d}{\sigma^2}\left[\frac{1 - e^{d\tau}}{1 - ge^{d\tau}}\right],$$

with

$$g = \frac{b_j - \rho\sigma\phi i + d}{b_j - \rho\sigma\phi i - d}$$

and

$$d = \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2(2u_j\phi i - \phi^2)}$$

As discussed further in Section 5, one may use the Fourier inversion method on the characteristic function to get the Heston model's log stock price pdf. One may also want to consider the impact of a given parameter on the pdf. Of particular interest is the volatility of volatility parameter, ξ, which impacts kurtosis. Increasing ξ increases both the peak and the thickness of the tails [17]. Greater kurtosis and the change in density means that the probability of more extreme prices are higher, an important consideration for any trader or market participant to assess risk. The impact of an increasing ξ can be seen in figure 2. Note that the author of this graph uses sigma, σ, to represent volatility of volatility.

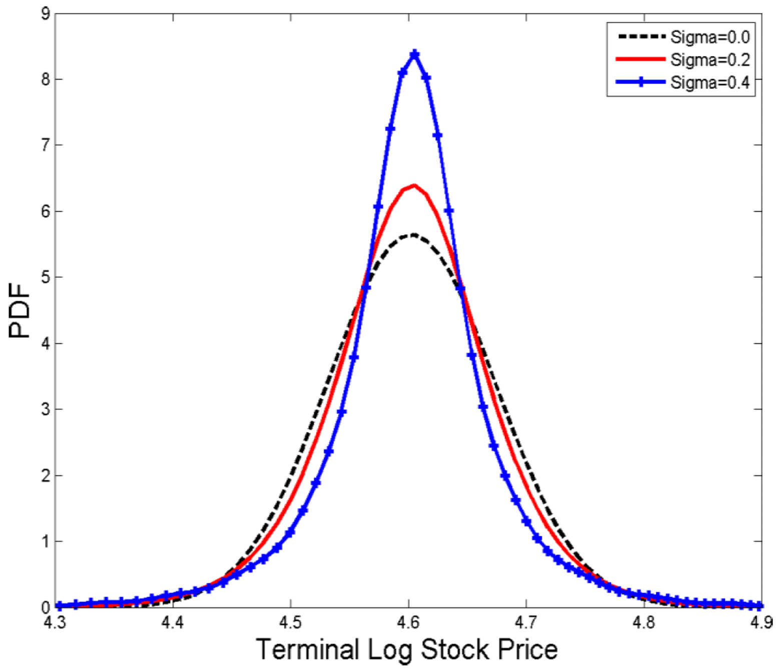

Figure 2: The effect of volatility of volatility on Heston log-price pdf [17]

Using the Gil-Pelaez (1951) inversion theorem, one can invert the characteristic function to solve for the in the money probabilities:

$$P_j(x, \nu, T; ln[K]) = \frac{1}{2} + \frac{1}{\pi} \int_0^{\infty} Re\left[\frac{exp(-i\phi ln[K]) f_j(x, \nu, T; \phi)}{i\phi}\right] d\phi$$

As we will explore in Section 5, one of the biggest drawbacks of the Heston model is its poor performance for pricing options with short maturities. A short-term implied volatility skew was one of the motivators to modify the Heston model to improve short-term pricing, which leads us to some of its derivatives, the Rough Heston model [18] and the Heston Model with Jumps.

4.2 The Heston Model with Jumps

Despite its popularity, the Heston Model is not without drawbacks. As the time to expiration decreases, the Heston model provides a poorer and poorer fit for the implied volatility of real world prices. To illustrate this fact, consider Figure 3 with the observed implied volatilities of real world bid and ask prices with the implied volatility smile of the Heston model superimposed.

In Figure 3, the triangles represent the bid and ask prices for SPX European options expiring on September 16th, 2005, as of close on September 15, 2005. The dashed line within the figure represents the implied volatility smile using parameters fitted from September 2005 options data. Note that the SPX was trading at $1227.73 as of close on September 15, 2005.

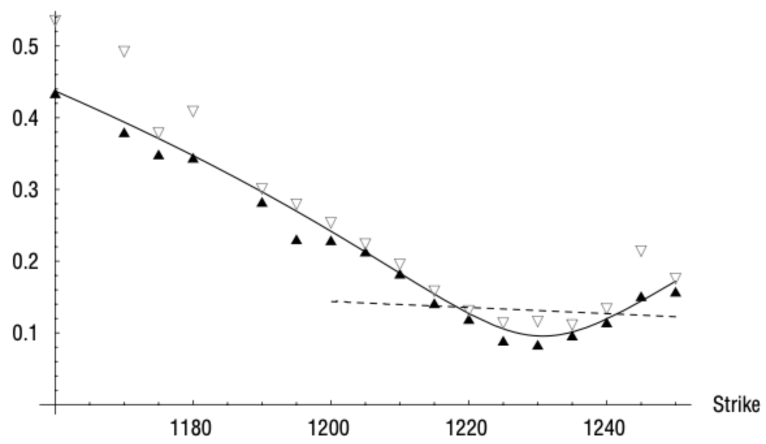

Figure 3: Observed Implied Volatility for 1 day to expiry Options with Heston Implied Volatility Smile Superimposed

Why is it the case that the Heston model fits real world data so poorly for options with short term maturities? In *The Volatility Surface: A Practitioners Guide* [22], Gatheral emphasizes that from the perspective of a trader, the explanation is rather simple; large moves sometimes occur, and it makes sense to bid for out-of-the-money options, which require extreme moves to end up in the money on very short time frames, in order to manage existing risk. For this reason, it makes sense to incorporate Jumps in the stock price process.

One such incorporation of a Jump diffusion model is the Heston Model with a Merton Jump Diffusion:

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t W_t^S + (e^{\alpha+\delta\epsilon} - 1) S_t dq$$

$$d\nu_t = \lambda(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^{\nu}$$

where $e^{\alpha+\delta\epsilon}$ is a log normally distributed Jump size with mean log jump α and standard deviation δ. $\epsilon \sim N(0,1)$ and dq is a Poisson process satisfying

$$dq = \begin{cases} 0 \text{ with probability } 1-\lambda_j dt \\ 1 \text{ with probability } \lambda_j dt \end{cases}$$

where λ_j is the jump intensity, sometimes called the Hazard rate. Furthermore we assume that the Brownian motion W_t^S and the Poisson Process d_q are independent of one another

[22].

The characteristic function of the Heston Model with Jumps can be shown to be:

$$f_j(X, \nu, t; \phi) = exp(C_j(T-t;\phi) + D_j(T-t;\phi)\nu + i\phi x + \psi(\phi)(T-t)$$

with

$\psi(\phi) = -\lambda_j i\phi(exp(\alpha + \delta^2/2) - 1) + \lambda_j(exp(i\phi\alpha - \phi^2\delta^2/2) - 1)$

and

$C_j(T-t;\phi)$, $D_j(T-t;\phi)$ as in Section 4.1 [22].

One can recognize that this particular incorporation of Merton Jumps into the Heston model's modifies the stock price process, but keeps the volatility process the same. As we will explore in Section 4.3, an alternative way to modify the Heston model is to instead consider a 'rough' volatility process while keeping the stock price process the same.

4.3 The Rough Heston Model

Proposed by El Euch and Rosenbaum in 2016 [11], then published in 2019, the Rough Heston Model is a modified version of the Heston model that instead considers rough volatility. From this, a question arises: what does 'rough' mean and how is it represented mathematically?

A generalization of Brownian Motion, called Fractional Brownian Motion (FBM), considers a variable $H \in (0, 1)$, known as the Hurst parameter, which describes the 'jaggedness' of the process. The definition of FBM includes a process, $\int_0^t (t-s)^{H-\frac{1}{2}} dW_s$, which is central to the dynamic of 'roughness' that was sought out to model volatility. This is introduced as $(t-s)^{\alpha-1}$ in the volatility process below, bringing the aspect of 'roughness' into the model [11]. Mathematically, the 'roughness' stems from the autocorrelation of the volatility process. That is, the correlation between the process across different time intervals. Although beyond this paper, stochastic processes which do not have independent intervals must be treated carefully as there are some properties that fail when the independence condition is not met.

Similar to the original Heston model, the Rough Heston model defines a stock price process that satisfies this SDE:

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t W_t^S$$

However, in the Rough Heston model, the volatility process $(\nu_t)_{0\leq t}$ is defined as the following:

$$\nu_t = \nu_0 + \frac{1}{\Gamma(\alpha)}\int_0^t (t-s)^{\alpha-1}\lambda(\theta - \nu_s)ds + \frac{1}{\Gamma(\alpha)}\int_0^t (t-s)^{\alpha-1}\xi\sqrt{\nu_s}dW_t^\nu$$

where the Hurst parameter $H = \alpha - \frac{1}{2}$

Another common and useful form of the stock price process is:

$$\frac{dS_t}{S_t} = \nu_t[\rho dW_t^\nu + \sqrt{1-\rho^2}dW_t^S]$$

where ν_t satisfies the same definition as above.

Similar to the Heston model, one can compute a quasi-closed form characteristic function for the log stock price. We will first let $X_t = log(S_t)$ and rewrite the equations above in the forward variance form. This requires some stochastic integrability conditions in a filtered probability space, however, this is beyond the scope of this project [18].

$$dX_t = \nu_t[\rho dW_t^\nu + \sqrt{1-\rho^2}dW_t^S] - \frac{1}{2}\nu_t^2 dt$$

and

$$\nu_t^2 = \psi(t) + \frac{\xi}{\Gamma(\alpha)}\int_0^t (t-s)^{\alpha-1}\nu_s ds$$

where $\psi(t)$ is the forward variance curve.

Then the characteristic function for the log stock price is:

$$\phi_t(T,a) = exp(iaX_t + \int_t^T D^\alpha h(a, T-s)\xi_t(s)ds)$$

where $\alpha = H + \frac{1}{2}$

and $h(a,t)$ is the solution of the fractional Riccati equation:

$$D^\alpha h(a,t) = -\frac{1}{2}a(a+i) + i\rho\gamma a h(a,t) + \frac{1}{2}\gamma^2 h^2(a,t)$$

with $I^{1-\alpha}h(a,0) = 0$ [19]

El Euch and Rosenbaum obtained this characteristic function by using the Fourier inversion method. A Fourier transform can be thought of as the 'link' between a characteristic equation and its density function. This is a common technique and it can be more efficient to use a characteristic equation in a Fourier space to compute option prices. Further, an algorithm known as the Fast Fourier Transform (FFT) can be implemented along with different option pricing models [21]. Of course, the Rough Heston method can still be quite computationally expensive, and this should be considered when determining if this model or certain numerical techniques are appropriate to use.

In their 2020 paper, Jeng and Kilicman [18] present an approximation formula to estimate the Rough Heston model. The approximation formula is obtained by using an expectation of the B-S solution and a second-order approximation of implied volatility. Then, the model parameters are calibrated by minimizing the squared error of implied volatility. The details and justification of convergence are beyond our scope, but are provided by Jeng and Kilicman [18]. They claim their approximation formula has greater simplicity and reduced computational complexity compared to the Fourier inversion method. In Figure 4, we present their comparison of implied volatility of the Rough Heston model under different moneyness and expiry conditions using different methods to solve Rough Heston.

In Figure 4, the solid lines were produced using the fractional Adams scheme, a well-known numerical method for solving fractional differential equations. On the other hand, the cross-marks are the results of the approximation method described above. As a reminder, moneyness describes the relationship between the underlying asset price at expiry and strike price. For a call option, moneyness can be summarized as:

- 'in-the-money' if $S_T > K$
- 'at-the money' if $S_T = K$
- 'out-the-money' if $S_T < K$

For put options, the conditions for 'in' and 'out-the-money' are swapped.

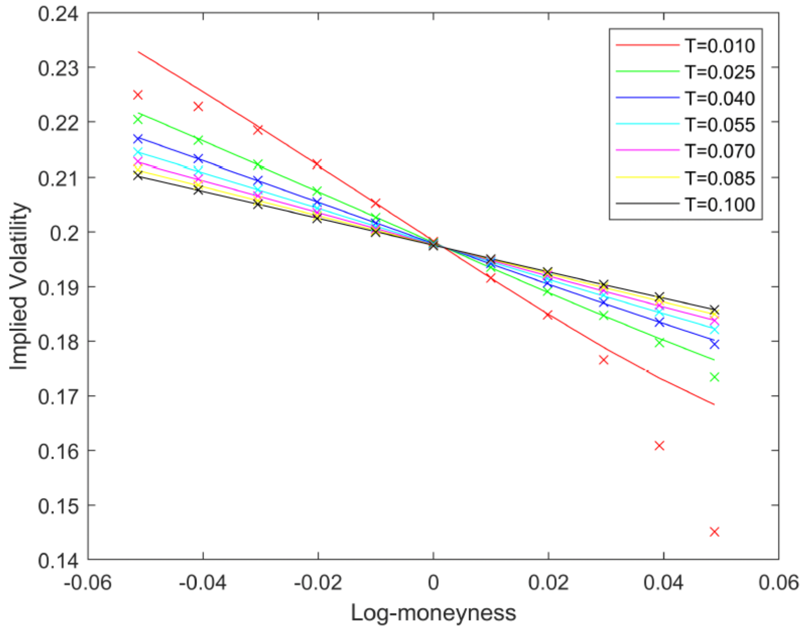

Figure 4: Implied volatility of Rough Heston for different expiries as a function of log-moneyness

4.4 Stochastic Volatility Model Calibration

Generally speaking, calibration allows for an understanding as to how accurately the predictive outputs of a model match observed data [33].

The act of calibration is a crucial process that each stochastic volatility model must undergo before it may be trusted to provide accurate option pricing outputs. The calibration process involves utilizing historical option data to optimize model parameters such as "initial and long-term volatility, mean-reversion speed, and correlation between asset price and volatility" [10] in order to ensure sufficient predictive accuracy of said model. These parameters must be optimized as they are not directly observable from option data, and thus may not be estimated [23]. The predictive accuracy of a stochastic volatility model may be maximized through various forms of calibration processes, which all attempt to minimize various types of error metrics [10]. Through our research, we found that these calibration techniques and error metrics vary from researcher to researcher.

In one example, Wu [10], optimizes their Heston model parameters using the built-in MATLAB non-linear least-squares solver, which minimizes the difference between the model's predicted option prices and the market valuation of the option contract. Once calibrated, Wu then utilizes these optimized Heston model parameters to predict option prices with confidence.

Similarly, Mrazek and Pospisil [23] utilize the same nonlinear least square Heston model calibration method as above, minimizing the option pricing error by finding the infimum of $G(\Theta)$, such that

$$G(\Theta) = \sum_{i=1}^{N} w_i |C_i^{\Theta}(t, S_t, T_i, K_i) - C_i^*(T_i, K_i)|^2$$

where:
$C_i^*(T_i, K_i)$ = actual option market price at time t
C^{Θ} = model option price
N = number of options used for calibration

In practice, Mrazek and Pospisil experimented with a variety of calibration methods, aiming to minimize both option pricing error and computational calibration time. They explain how utilizing approximation instead of an exact formula during the calibration process considerably increases computational efficiency, concluding that a combination of GA, a generic local minimizer function [34], and lsqnonlin MATLAB optimization along with approximation techniques led to precise parameter optimization while still maintaining computational efficiency.

In another example, Jin [6] minimizes Implied Volatility Root Mean Square Error (IVRMSE), an error metric which is a measure of the distance between real market and model- obtained implied volatility. Her optimized model parameters are a function of:

$$\hat{\theta} = \text{argmin}_{\theta} \sqrt{\frac{\sum_{i=1}^{N} (IV_i - IV_i^{\theta})^2}{N}}$$

where:
$\hat{\theta}$ = array of optimized model parameters
θ = array of model parameters
IV_i = implied volatility calculated from market option value
IV_i^{θ} = implied volatility calculated from the option value output of the model
N = number of options

Once optimized, minimized IVRMSE is obtained by:

$$IVRMSE(\hat{\theta}) = \sqrt{\frac{\sum_{i=1}^{N} (IV_i - IV_i^{\theta})^2}{N}}$$

Where $\hat{\theta}$ calculated prior is inserted into the above equation.

Here, Jin demonstrates that not only is calibration useful to obtain optimized model parameters, but comparing IVRMSE between models allows for a direct comparison of model performance relative to the input data. Specifically, Jin ordered her model performance as such: Rough Heston with Jumps, Heston with Jumps, Rough Heston, and Heston, (ranked from best performance to worst) once having calibrated each of these models.

We will investigate further comparisons of option pricing models in the following sections.

5 Model Analysis and Empirical Results

In this section, we investigate the differences between the Black-Scholes, Heston, Heston with Jumps, and Rough Heston models by following the progression of their development. We take particular interest in the behaviour of volatility, the impact of different volatility definitions on probability density functions of stock prices, and the different time to maturity and moneyness conditions under which each model performs best.

5.1 Black-Scholes vs Heston

To compare the Black-Scholes and Heston models, we begin by comparing their definitions of the stock price process $(S_t)_{0 \leq t}$. As described in Section 4.1, the Heston model has a similar stock price definition, where the major difference lies in the definition of the volatility process: constant or stochastic. Considering Heston's definition, but with a volatility process that happens to be constant, one can show that the B-S model is actually a special case of the Heston model [7]. This process involves variable substitution and manipulating the characteristic function to reach the B-S formula. One may also obtain the pdfs of the log stock returns by using the inverse Fourier transform on their characteristic equation. Compared to empirical data, the Heston model pdf is much more accurate than the B-S pdf [15]. It should be noted that the Heston pdf has 'fatter' tails, or greater kurtosis, than the B-S pdf, which is part of what improves its alignment to empirical returns. This is displayed in Figure 5 below where 1) the empirical log return probability density was estimated using the technique of kernel density estimation, 2) the B-S log return is described by the log of the normal density using implied volatility, and 3) the Heston log-return pdf was also estimated using kernel density with the Euler method for simulation. These densities are centered on r. It should be noted that one may instead use a Fourier transform on the log-price characteristic function to reach the Heston log-price pdf estimate. Bucic, the author of this graph, notes that the Fourier transform technique performed poorly for S&P 500 daily return data and thus used the kernel density technique [15].

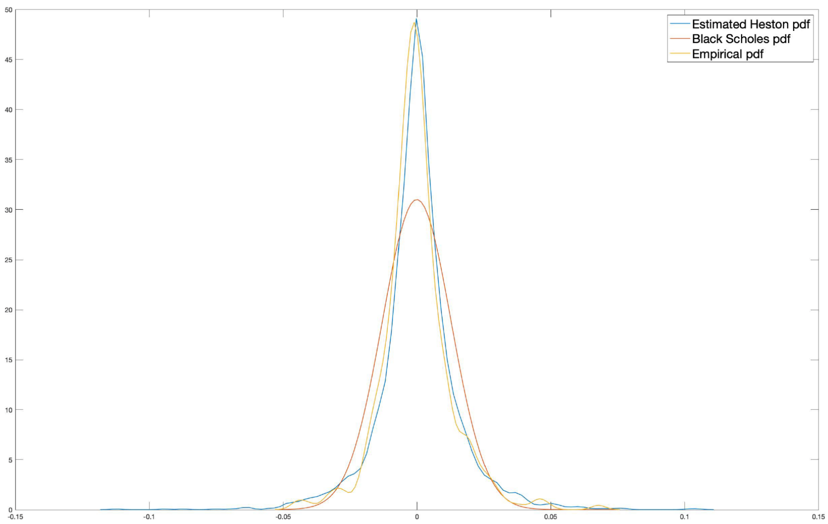

Figure 5: Comparison of estimated log-return pdfs of empirical data, B-S model, and Heston model

The volatility of volatility parameter, ξ, affects the peak and tails of the pdf. As ξ increases, so does the peak of the pdf and creates heavier tails on either side [17].

To demonstrate how the Black Scholes and Heston models compare in their accuracy in different market situations, we will present results based on Mean Absolute Percentage Error for different time horzizons and moneyness. These results are summarized in Table 1. Chakrabarti and Santra (2017) compared historical S&P 500 options data across different combinations of time to expiry, which we may refer to as time horizons, and % moneyness, which is the percentage difference between the current price S and strike price K:

$$\%Moneyness = \frac{S-K}{K}$$

This study uses the following definitions for different ranges of $\%Moneyness$:

- Deep out the Money (DOTM): lower than -6%
- Out the Money (OTM): -2% to -6%
- At the Money (ATM): -2% to 2%
- In the Money (ITM): 2% to 6%
- Deep in the Money (DITM): greater than 6%

Before stating their results, it is important to note that the Heston model used in this study was calibrated using the non-linear least square method by using a built-in solver in MATLAB [16]. Calibration is the process of determining the set of parameters that minimize the difference between the model prices and true market prices. In particular, non-linear least squares estimation is an optimization problem that estimates the value of a parameter that minimizes the sum of squared deviations from the observed data. For non-linear functions like the ones we are investigating, there is an added cost of greater numerical computation and it is unlikely that an analytical result can be achieved [20].

To compare the accuracy of each model price to the real market price, Mean Absolute Percentage Error (MAPE) was used.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|model\ \ price - market\ \ price|}{market\ \ price}$$

Further, implied volatilities were computed using another MATLAB built-in function. In Table 1, we summarize the results by indicating which model provided the most accurate result under different moneyness and time horizons.

Time to Maturity/Moneyness	DOTM	OTM	ATM	ITM	DITM
≤ 45 days	Heston	Heston	B-S	B-S	B-S
$45 - 90$ days	Heston	Heston	Heston	B-S	B-S
≥ 90 days	Heston	Heston	Heston	Heston	Heston

Table 1: B-S vs Heston: Best Performer by MAPE

Overall, B-S performs better than Heston under short time horizons (≤ 45 days) compared to the Heston model, which is much more accurate than B-S over longer time horizons. This result is reached because the Heston model does not well estimate high volatility in the short term and the B-S assumption of constant volatility doesn't hold over longer intervals. In most scenarios, the Heston model suggests lower volatility than is true for the market. Interestingly, both models perform very poorly for Deep in the Money (DITM) options, however, both improve as the time horizon increases, with the Heston model becoming more accurate more quickly.

5.2 Heston vs Heston with Jumps

The primary motivation to add Merton Jump Diffusion to the Heston model is Heston's poor performance for short expirations. So, we will focus on investigating the impact on volatility skews and surfaces of adding jumps to the Heston model over short time horizons. Further, Gatheral shows that adding jumps does very little to impact the shape of the volatility surface for options with longer expirations [22].

Different calibrations of the Heston Model with Jumps can be obtained, and Gatheral fits the parameters to the volatility surface for SPX on September 15, 2005. Heston with Jumps results in an implied volatility surface that holds the same significant features of the empirical volatility surface. Most importantly, it has this behaviour over short time horizons without losing it over the long-term. In the below Figures 6 and 7, the lighter shaded surface represents the same empirical volatility surface for September 15, 2005 SPX. The darker surfaces are the implied volatility surfaces of the Classical Heston and Heston with Jump models, respectively. These implied volatility surfaces have been shifted down slightly for readability, so the true surfaces are more accurate than represented. One can verify the claim that adding jumps to the Heston model improves the accuracy of short-term implied volatility by observing Figures 6 and 7.

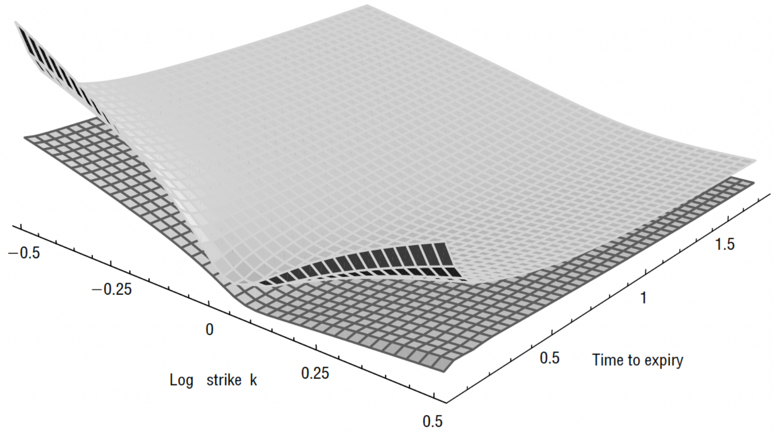

Figure 6: Empirical SPX Volatility Surface vs Implied Volatility Surface of the Classical Heston Model [22]

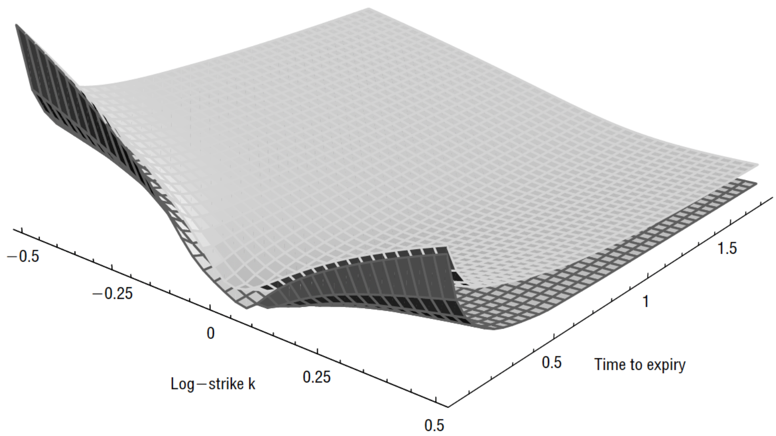

Figure 7: Empirical SPX Volatility Surface vs Implied Volatility Surface of Heston Model with Jumps [22]

As a result, one can expect more accurate short-term results by adding jumps to the Heston model. Additionally, jumps can be added to the volatility process as well, which Gatheral concludes is less realistic to use in practice as the increased complexity of calibration outweighs the improved fit of the volatility surface. The addition of another volatility jump is beyond this project, but this is expanded on further by Gatheral [22].

5.3 Heston vs Rough Heston

To compare the Heston and Rough Heston models, we will focus on their volatility processes and implied volatility results over short time horizons. As in the case of adding jumps to Heston, the modification of rough volatility does not have a meaningful impact on the accuracy of the model for long time horizons.

Before diving into the market scenarios, one may be curious how the definitions of Heston and Rough Heston are further related. By substituting the roughness index $\alpha = 1$, or equivalently the Hurst parameter $H = \frac{1}{2}$, into the Rough Heston volatility process, one can easily show that the original Heston volatility process is the result. So, the Heston model is a special case of the Rough Heston model [6].

Given Heston's poor performance pricing options over short time periods, the results of the Rough Heston model and its volatility under these conditions are of particular interest. El Euch and Rosenbaum conducted a numerical experiment to explore exactly this volatility behaviour [11]. In their numerical scheme to approximate the log stock price characteristic function, a trapezoidal discretization of the implied Volterra integral equation is used. The result is an implicit scheme and so a pre-estimation of the next result is used. It is shown that the method converges theoretically with an error proportional to the size of 1 time-step in the defined discrete-time grid. Then, by applying this method to a classical Heston, where $\alpha = 1$, and Rough Heston model, with $\alpha = 0.6$ or equivalently $H = 0.1$, option prices can be obtained. El Euch and Rosenbaum computed the at-the-money skew, which is the derivative of the implied volatility with respect to log stock price, and plotted this against time to maturity for both classical and Rough Heston as described above. In Figure 8, it is clear that the skew of the Rough Heston model becomes very large as maturity approaches zero. This is an advantage of the Rough Heston model because this explosion much more accurately reflects empirical volatility from market data [18].

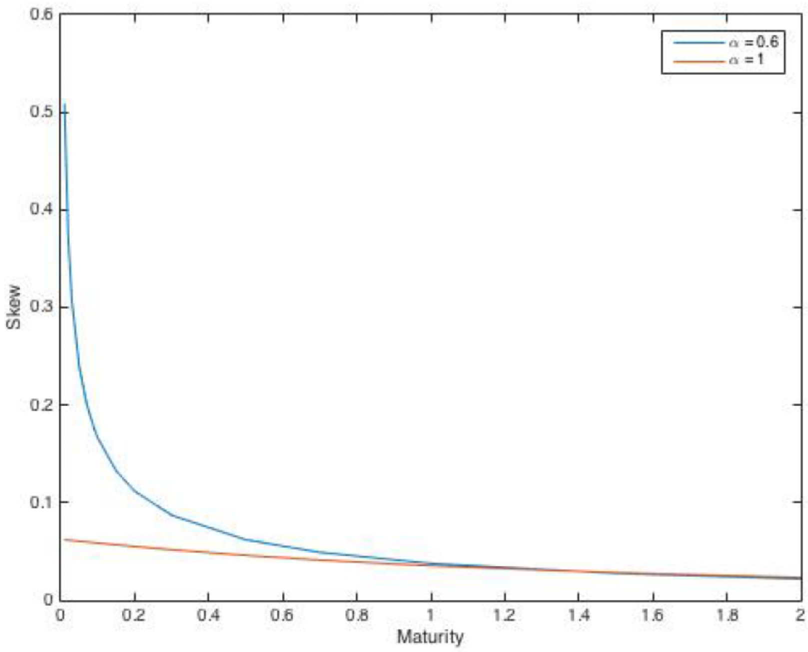

Figure 8: At-the-money volatility skew for Heston ($\alpha = 1$) and Rough Heston ($\alpha = 0.6$) models [11]

Furthermore, Friz and Keller-Ressel [24] describe a moment explosion within a stochastic volatility model as the moment the asset price S_t becomes infinite after some finite amount of time. Moment explosions are important to consider, as the implied volatility surface of a stochastic model is heavily influenced by the smallest and largest finite moments of said model. These moments are inverse functions of moment explosion time of S_T, where T = maturity time of the asset [27][25] as cited in [24]. Comparing moment explosion times, Majid and Keller-Ressel [26] describe their research results through utilization of the asymptotic implied variance slope (AIVS). This slope, defined as:

$$AIVS^{\pm}(T) = \text{limsup}_{x \to \pm\infty} \sigma^2_{IV}(T, x)/|x|$$

where:
$x = \log(K/S_0)$ or log-moneyness
where $K =$ option strike price, $S_0 =$ underlying asset price at time 0
$T =$ time to maturity
$\sigma^2_{IV}(T, x) =$ Black-Scholes implied volatility

allows for two important conclusions to be described. Firstly, for short maturity and low strike price options, the implied volatility curve of the Rough Heston model is much steeper than the non-rough Heston model, which follows the results demonstrated in Figure 8, confirming again how adding roughness to the Heston model leads to a more accurate representation of the volatility behavior found within real-world options data [18]. Secondly, the Rough Heston implied volatility curve for short maturity, high strike price options explodes at the same power-law rate as the low strike price curve [26], which as described above, infers that the Rough Heston model more accurately reflect real-world market data volatility when pricing these specific options with high strike price, as compared to the non-rough Heston model.

5.4 Empirical Results

In this section, we utilize Python scripts and historical option data to present relevant empirical results. These results are our attempt to personally confirm and supplement our expository findings, focusing on the Black-Scholes model and the behavior of call option prices and implied volatility.

Firstly, we ordered and downloaded 2023 first quarter End-Of-Day SPX option chains from optionsDX [28], a free to use options database website. SPX option chains are composed of historical call and put option contracts with Standard and Poor's 500 index as the underlying asset. Consistent quote and expiry dates were utilized to limit the number of independent variables, thus we pulled options with March 1st, 2023 and June 16th, 2023 quote and end dates, respectively. These dates were chosen randomly.

Within our Python script, we imported these options contracts and assigned them to a dataframe along with their associated quote dates, expiry dates, the price of their underlying S&P index, call bid and ask prices, and strike price. We then dropped blank values from the dataframe and calculated the time to expiry for each contract in years as $\frac{\text{Expiry Date - Quote Date}}{365}$).

Next we calculated the implied volatility from the Black Scholes Model of each option in our chosen data using the root finding Newton Raphson Method.

The Newton Raphson method requires $f(\sigma)$ to be differentiable and $f'(\sigma) \neq 0$

As defined in section 3, the implied volatility function is

$$f(\sigma) = S_0 N(d_1(\sigma)) - Ke^{-r(T-t)} N(d_2(\sigma)) - C$$

where C is the observed option price $N(x)$ is the cumulative normal distribution $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-z^2/2}\, dz$

and

$$d_1 = d_+ = \frac{1}{\sigma\sqrt{T-t}}(ln(\frac{S_t}{K}) + (r + \frac{1}{2}\sigma^2)(T-t))$$

$$d_2 = d_- = \frac{1}{\sigma\sqrt{T-t}}(ln(\frac{S_t}{K}) + (r - \frac{1}{2}\sigma^2)(T-t)) = d_1 - \sigma\sqrt{T-t}$$

then it can be shown by [37] that

$$f'(\sigma) = S_0 * \sqrt{T-t} * N'(d_1)$$

Where $N'(x)$ is the pdf of the standard normal distribution $\frac{e^{-x^2/2}}{\sqrt{2\pi}}$ and $t = 0$.

By the Newton Raphson Method, we first set $f(\sigma) = 0$ then chose an initial guess for σ, $\sigma_0 = 0.3$ for each option in the data set.

Assuming σ_0 is close to σ the desired root of $f(\sigma) = 0$, we approximate $y = f(\sigma)$ near its root by evaluating the root of the tangent line formed at $(\sigma_0, f(\sigma_0))$. We then use the root of this tangent line to approximate σ again and call this approximation σ_1. Repeating this process leads to the iterative equation [36]:

$$\sigma_{n+1} = \sigma_n - \frac{f(\sigma_n)}{f'(\sigma_n)}$$

The sequence of approximations will converge rapidly towards σ. Using data frames in Python we are able to calculate each σ_n at each step in a for loop. For our calculations we set the break condition for the loop to be when

$$\sum_{i=1}^{M}(|\sigma_{i,n+1} - \sigma_{i,n}|) < 10^{-10}$$

Where $M = 256$ is the number of option quotes in our data set and $\sigma_{i,n}$ is the n^{th} approximation of the i^{th} option in our data set.

The Newton Raphson Method above thus provided us with the resulting implied volatility smile of our observed option set in 9. The accompanying Python code can be found in Appendix D.

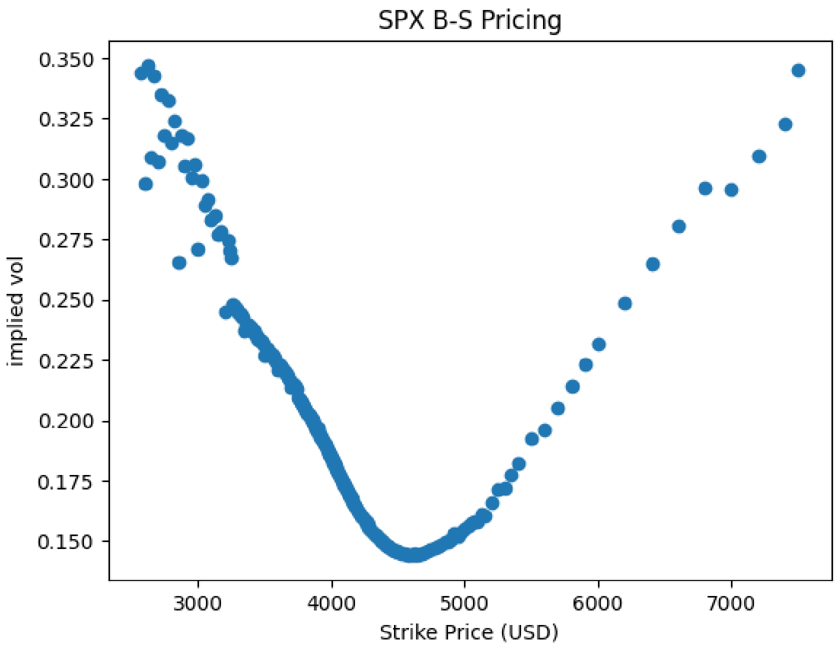

Figure 9: Black-Scholes Implied Volatility vs Strike Price of SPX European Call Options expiring on 2023-06-16 as of close on 2023-03-01 where S = 3952.01 and r = 0.041.

Next, we created a secondary Python script with a similar yet independent process as the Newton Raphson code described above. In this case, the purpose of our script was to calculate Black-Scholes option prices with constant volatility, a key assumption and input of this famous model. Our particular goal was to observe the relationship between the varying option strike prices within our dataset, and the call and put option prices the model outputs. Here we utilized the same SPX dataset and import/data cleaning methods as described on page 20, and organized this SPX data into dataframes for simple manipulation [29]. For each respective option contract within our dataframe, following a process similar to the methods described by CodeArmo [31] but converting the function to accomodate dataframe inputs, we first calculated d_1 and d_2, where

$$d_1 = \frac{ln\frac{S}{K} + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

and

$$d_2 = d_1 - \sigma\sqrt{T}$$

where:
S = underlying S&P index price
K = Strike price
r = risk-free rate (held constant at 0.01)
σ = volatility (held constant at 0.3)
T = time to expiry in years

Next, for each individual contract we used the equations:

$$BS_C = S * N(d_1) - K^{-rT} * N(d_2)$$

$$BS_P = K^{-rT} * N(d_2) - S * N(d_1)$$

BS_C = Black-Scholes call option price
BS_P = Black-Scholes put option price
N = CDF of normal distribution (utilizing the scipy Python package)

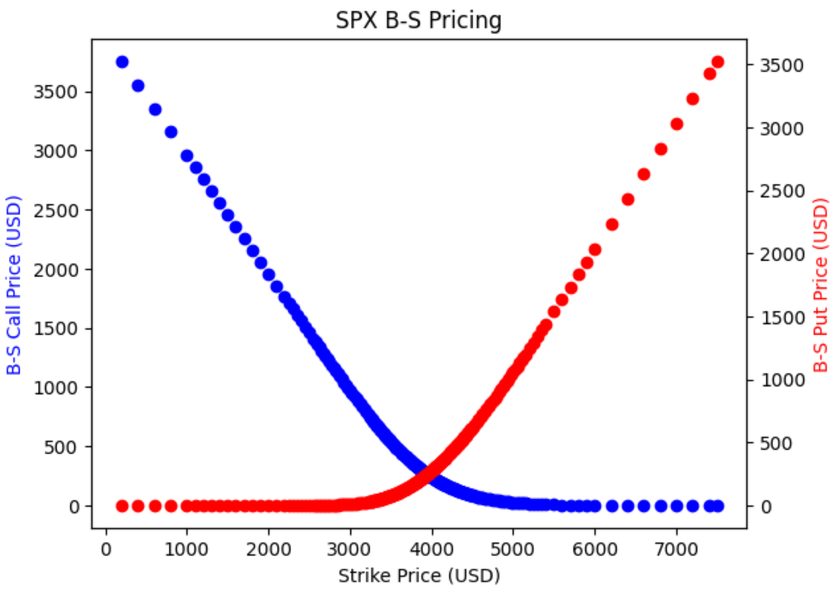

Figure 10: Black-Scholes put and call SPX option price vs. strike price

Created using Matplotlib [30], Figure 10 visualizes the output of the code above. This chart describes the relationship between our Black-Scholes model output call and put option prices and the strike price of each option. Taking note that the underlying S&P500 index price was consistent at $3952 USD throughout the dataset, by [32], the more 'in-the-money' a call option is, the further the strike price sits below the underlying S&P index price. Similarly, the higher the strike price sits above the underlying S&P index price, the more 'in-the-money' a put option is. The more 'in-the-money' these options are, the higher the potential payoffs are for the option contract holder, and thus the option contract should be costlier to purchase. 'out-the-money' options, or those which will not be exercised unless the underlying S&P index price shifts into the money, hold significantly less value than 'in-the-money' options, and should be priced as such. Figure 10 demonstrates that our Black-Scholes option pricing model reflects these trends. The Python code associated with this Black-Scholes model can be found in Appendix E.

Further, we explored built-in simulation functionality in MATLAB. There exist default objects for Geometric Brownian, Heston, and Rough Heston in the Financial Toolbox [12], a package available from MathWorks that includes tools to price financial instruments and model stochastic differential equations. In Figure 11, we used the built-in object, 'gbm', to simulate a stock price that follows a Geometric Brownian Motion, as in the Black-Scholes Model. This object takes at least two arguments; the required arguments are the return, μ, and instantaneous volatility, σ. Either of these arguments can be a constant real number, a matrix of volatility states, or a deterministic function of time. Other properties of the object include the start time and state, the simulation method, and correlation, drift, and diffusion parameters that impact the SDE. In Figure 11, we set the parameters: return $\mu = 0.1$, volatility $\sigma = 0.05$, 'StartTime' = 0, and the initial stock price 'StartState' = 1. The partition over time was chosen so that the final time was $T = 1$. With these parameters, we used the default 'simulate' functionality and used a simple for loop to execute 100 of these simulations and graphed their results. The accompanying code may be found in Appendix A.

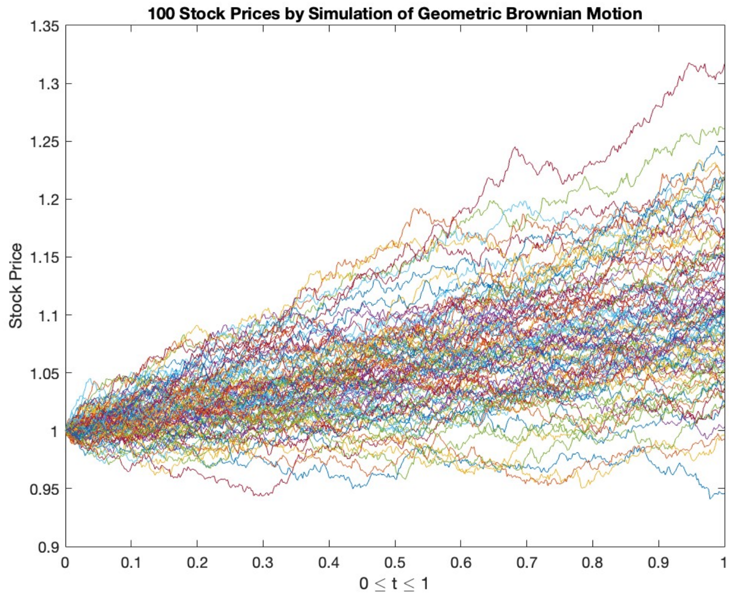

Figure 11: MATLAB Simulation of 100 Stock Prices using Geometric Brownian Motion

The built-in 'heston' object can be used to simulate stock prices as defined by the Heston model. The object defines an SDE for the stock price similar to that of the 'gbm' object, however, it of course defines volatility as a stochastic process. This volatility process is modeled by a mean-reverting Cox-Ingersoll-Ross (CIR) process, as described in Section 4.1. In addition to the return μ and instantaneous volatility σ arguments, the 'heston' object also requires 'level' and 'speed'. Where 'level' represents the mean-reversion level and 'speed' is the mean-reversion speed of the volatility process [13]. These parameters are denoted θ and λ respectively in our Heston model in Section 4.1. We again used the 'simulate' function to simulate and graph 100 stock prices under the Heston model. We used the same parameters as our simulation of Geometric Brownian Motion with additional parameters, 'level' $\theta = 0.1$ and 'speed' $\lambda = 0.2$. We graphed these simulations in Figure 12 and the accompanying code may be found in Appendix B.

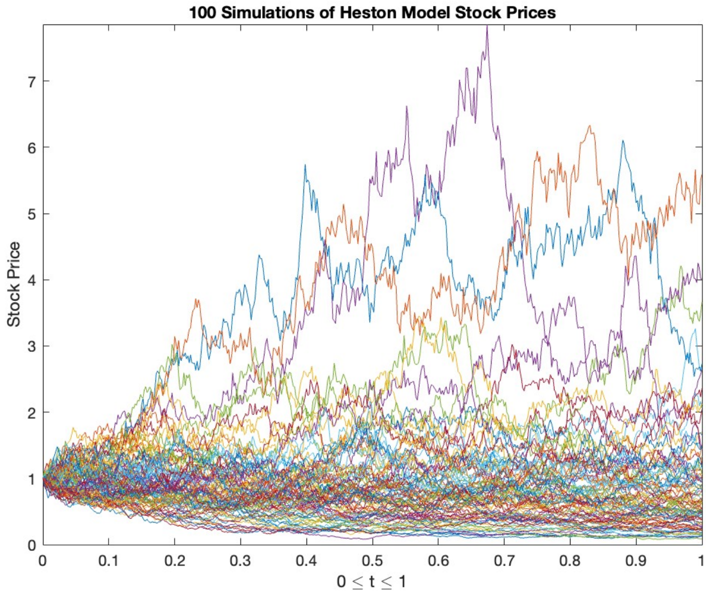

Figure 12: MATLAB Simulation of 100 Stock Prices under the Heston Model

The 'roughheston' object is a recent addition to MATLAB and is implemented very similarly to the 'heston' object. The primary difference is that a Fractional Brownian Motion (FBM) is incorporated into the CIR volatility process. With the addition of the FBM, there is another required parameter to define a 'roughheston' object, a roughness index [14], α, defined similarly as in Section 4.3, however, in MATLAB, we face the restriction $-0.5 < \alpha < 0$. Where the Hurst parameter, H, is instead obtained by $H = \alpha + \frac{1}{2}$ to be used in the FBM [14]. We again simulated stock prices using the same set of parameters, plus the roughness index $\alpha = -0.3$ for the 'roughheston' object. Graphed results of these simulated prices can be observed in Figure 13 and the corresponding MATLAB code may be found in Appendix C.

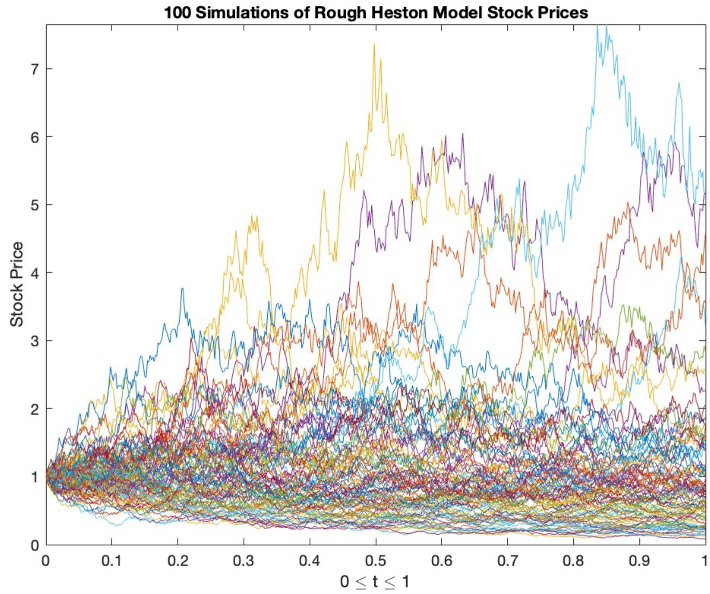

Figure 13: Simulation of 100 Stock Prices under the Rough Heston Model with $H = 0.2$

6 Conclusion

In this project, we investigated different methods to price an important class of financial derivatives, European options. We followed the path of historical development of these models to best identify the issues they experience, and how new models proposed a mathematical basis for the changes to address these issues. We presented how stochastic volatility considerations, as well as the additions of jumps and roughness, can improve the accuracy of implied volatility behaviour relative to true market volatility. However, one of the biggest drawbacks of stochastic volatility models is an increase in mathematical, computational, and calibration complexity.

The Black-Scholes model is popular and relatively simple, quick to compute, and performs well for pricing options close to expiry. However, its assumption that volatility is constant is unrealistic and results in poor performance when pricing longer term options. Through our implementation of the Newton-Raphson method, we plotted the resulting implied volatility smile created by the Black-Scholes model in Figure 9, which demonstrates this inaccuracy. The shift to stochastic volatility in the Heston model aimed to address these issues, and was, in general, successful. This dynamic volatility better reflects market characteristics and the resulting asset price more accurately reflects the empirical prices, as demonstrated in Figure 5. This graph shows the probability densities of log stock returns of the Black-Scholes and Heston models compared to empirical data. This change results in improved long-term pricing by the Heston model, however, the cost of this improvement is the accuracy of short-term pricing and volatility. This is attributed to the fact that the Heston model struggles to capture high short-term volatility due to the mean reverting property of the volatility process. Additionally, there is a significant increase in the complexity of calibration and computational demand when utilizing the Heston model.

For markets exhibiting high short-term volatility, one may consider one of two popular modifications to the Heston model, both designed to address the issue of short-term accuaracy and to better capture skew behaviour. The first is adding 'jumps' to the price process, and the other is adding 'roughness' to the volatility process. A 'jump' is a random process with a log-normal distribution which is added to the Heston stock price process. This accounts for sudden movements or changes in the stock price which may have otherwise been missed. Another modification is to make the volatility process 'rough', which is based on the autocorrelation of the process. This allows one to consider more jagged volatility behaviour, which can more accurately reflect sharp changes in the market. We compared how the characteristic functions and implied volatility of these stochastic volatility models behave, particularly over short maturities. A key take-away is the volatility curves of the modified models in Figures 7 and 8 more accurately reflect high short-term volatility which the Heston model fails to capture in Figure 6. Either of these 'roughness' or 'jumps' modifications to the Heston model improves short-term accuracy without sacrificing long-term performance. However, trade-offs exist when implementing these variations to the Heston model, in the form of even greater complexity of calibration and computation.

Through our own simulations, we learned that implementation can be tricky even with simpler models. Firstly, we used built-in functions in MATLAB to simulate the random behaviour of stock prices using Geometric Brownian Motion, and under the Heston and Rough Heston models. These functions were quite accessible and easy to use, as quality documentation is readily available. Furthermore, we implemented the Black-Scholes model to price European call and put options using publicly available SPX options data. This was a more challenging task to complete as it was difficult to manipulate our data and model in order to obtain meaningful output.

Beyond the scope of this project are other advancements to option pricing models, including further modifications to the Heston model, such as Jin's Rough Heston with Jumps [6] and Quadratic Rough Heston [35]. Next steps which could be taken would be further investigation into these new option pricing models, exploring both calibration and numerical methods at a deeper level mathematically, and the use of Monte Carlo and quasi-Monte Carlo methods used for option pricing simulations. Further, we could expand our scope to include more complex financial derivatives beyond European options.

With the continued growth of the financial derivatives market and the volatile conditions today, it is incredibly important to understand how different option models perform. By understanding trade-offs between computational complexity and the accuracy of pricing and volatility, we can make an informed decision about which model to use in a given context. In this project, we have explored four option pricing models and how they compare. We concluded when each is best to use depending on market conditions and the purpose of the user. As new models and modifications have already been proposed in the last decade, we can expect financial mathematics to continue to develop.

References

[1] P.Boyle, F.Boyle, *Derivatives: The Tools That Changed Finance*, Chapters 1, 2, and 5, Risk Books, London, United Kingdom (2001). ISBN: 9781899332885

[2] D.J. Higham, *An Introduction to Financial Option Valuation: Mathematics, Stochastics, and Computation* Chapter 1, 2, Cambridge University Press, New York, USA (2004). ISBN: 9780521838849

[3] M. Capiński, E. Kopp, J. Traple, *Stochastic Calculus for Finance*, Mastering Mathematical Finance, Chapter 2, 3, Cambridge University Press, Cambridge, United Kingdom (2012). ISBN: 9781139017367

[4] F. Black, M. Scholes, *The Pricing of Options and Corporate Liabilities*, The Journal of Political Economy, Vol.81 No.3, p.637-654, The University of Chicago Press, Chicago, USA (1973). DOI:10.1086/260062

[5] E. Derman, M.B. Miller, *The volatility smile : an introduction for students and practitioners*, 1st edition, Chapter 8, John Wiley & Sons, Inc., Hoboken, New Jersey, USA, (2016). ISBN: 9781118959169

[6] Y. Jin, *Option Pricing Under Rough Heston Model with Jumps*, University of Calgary, Calgary, Canada (2022). DOI:10.11575/PRISM/40309

[7] F.D. Rouah, *The Heston Model and Its Extensions In Matlab and C#*, Chapter 1, John Wiley & Sons, Inc., Hoboken, New Jersey, USA, (2013). ISBN: 9781118695173

[8] S.L. Heston, *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies, Vol.6 No.2, p.327-343, Oxford University Press, Oxford, United Kingdom (1993). DOI:10.1093/6.2.327

[9] G. Iyer, *Stochastic Calculus for Finance: Brief Lecture Notes*, Chapter 4, Carnegie Mellon University, (2020), Accessed March 6, 2025, URL: https://www.math.cmu.edu/~gautam/sj/teaching/2021-22/944-scalc-finance1/

[10] H.F. Wu, *From Constant to Stochastic Volatilty: Black-Scholes Versus Heston Option Pricing Models*, Senior Projects Spring 2019, Bard College, (2019). Accessed March 6, 2025, URL: https://digitalcommons.bard.edu/senproj_s2019/163

[11] O. El Euch, M. Rosenbaum, *The characteristic function of rough Heston models*, Mathematical Finance, vol. 29, no. 1, pp. 3–38, Wiley Periodicals, Inc., (2019). DOI:10.1111.12.17

[12] *gbm Geometric Brownian motion (GBM) model*, MathWorks: MATLAB Help Center, Accessed March 8, 2025, URL: https://www.mathworks.com/help/finance/gbm.html

[13] *heston Heston model*, MathWorks: MATLAB Help Center, Accessed March 28, 2025, URL: https://www.mathworks.com/help/finance/heston.html

[14] *roughheston Rough Heston Model*, MathWorks: MATLAB Help Center, Accessed March 28, 2025, URL: https://www.mathworks.com/help/finance/roughheston.html

[15] I. Bucić, *Heston vs Black Scholes stock price modelling*, Degree Project, Linnaeus University, Sweden, (2021). Accessed March 10, 2025, URL: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1576236&dswid=-123, DiVA ID:diva2:1576236

[16] B.B. Chakrabarti, A. Santra, *Comparison of Black Scholes and Heston Models for Pricing Index Options*, Working Paper Series, No. 796, Indian Institute of Management Calcutta, Kolkata, India, (2017). Accessed March 10, 2025, URL: https://www.iimcal.ac.in/faculty/publications/working-papers/archive/2017

[17] T.T. Suki, A.B.M. Shahadat Hossain, *A Comparative Analysis of the Black-Scholes-Merton Model and the Heston Stochastic Volatility Model*, GANIT Journal of Bangladesh Mathematical Society, Vol. 39, pp. 127-140, Dhaka, Bangladesh, (2019). DOI:10.3329/ganit.v39i0.44168

[18] S.W. Jeng, A. Kilicman, *Approximation Formula for Option Prices under Rough Heston Model and Short-Time Implied Volatility Behavior*, Symmetry, Vol. 12, No. 11, p. 1878-, MDPI, Universiti Putra Malaysia, Seri Kembangan, Malaysia, (2020). DOI:10.3390/sym12111878

[19] O. El Euch, J. Gatheral, M. Rosenbaum, *Roughening Heston*, Risk, May 2019, pp.84-89, (2019). DOI:10.2139/ssrn.3116887

[20] *Least Squares*, National Institute of Standards and Technology, Accessed March 17, 2025, URL: https://www.itl.nist.gov/div898/handbook/pmd/section4/pmd431.htm

[21] *Linking Fourier Inversion Methods to Option Pricing*, Medium, February 2, 2024, Accessed March 17, 2015, URL: https://medium.com/@alexanderjdyson/linking-fourier-inversion-methods-to-option-pricing-3825f46f3239

[22] J.Gatheral, *The Volatility Surface: A Practitioner's Guide*, John Wiley & Sons, Inc., September 2006, Hoboken, New Jersey, USA, (2006). ISBN: 9780471792512

[23] M. Mrazek, J. Pospisil, *Calibration and simulation of Heston model* De Gruyter Open Access, University of West Bohemia, Plzen, Czech Republic, May 2017. DOI:10.1515/math-2017-0058

[24] M. Keller-Ressel, P.Friz, *Moment explosions in stochastic volatility models*, January, 2007. URL:https://www.researchgate.net/publication/228837899_Moment_explosions_in_stochastic_volatility_models

[25] S.Benaim, P.Friz, *Smile Asymptotics II: Models with Known Moment Generating Function*, Aug 2006, URL:https://arxiv.org/abs/math/0608619

[26] M. Keller-Ressel, A. Majid, *A comparison principle between rough and non-rough Heston models - with applications to the volatility surface*, Quantitative Finance, Volume 20, Issue 6, p.919-933, February 2020. DOI:10.1080/14697688.2020.1714702

[27] R. Lee, *the Moment Formula for Implied Volatility at Extreme Strikes*, Chicago University, August 11, 2003, Accessed March 31, 2025, URL: https://math.uchicago.edu/~rl/moment.pdf

[28] *Historical Intraday quotes and data on US and Cryptocurrency options for common exchange traded securities including the SPX, VIX, SPY, and Bitcoin (BTC)*, optionsDX, 2025, Accessed February 26, 2025, URL: https://www.optionsdx.com/

[29] *pandas.Dataframe*, Pandas, Accessed March 10, 2025, URL:https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

[30] *Using Matplotlib*, Matplotlib, Accessed March 12, 2025, URL: https://matplotlib.org/stable/users/index.html

[31] *Black Scholes Model Python*, CodeArmo, December 2022, Accessed March 1, 2025, URL: https://www.codearmo.com/python-tutorial/options-trading-black-scholes-model

[32] *The Benefits of 'In The Money Calls'*, Investopedia, May 2024, Accessed March 25, 2025, URL: https://www.investopedia.com/ask/answers/042215/when-call-option-considered-be-money.asp

[33] *Model Calibration, Explained*, Medium, Jan 2025, Accessed March 27, 2025, URL:https://medium.com/data-science/model-calibration-explained-a-visual-guide-with-code-examples-for-beginners-55f368bafe72

[34] *GA*,MATLAB Help Center, Accessed April 8, 2025, URL: https://www.mathworks.com/help/gads/ga.html

[35] J. Gatheral, P. Jusselin, M. Rosenbaum, *The quadratic rough Heston model and the joint S&P 500/VIX smile calibration problem*, January 8, 2020. DOI:10.48550/arXiv.2001.01789

[36] S. Akram, QU. Ann, *Newton Raphson Method*, International Journal of Scientific & Engineering Research, Vol.6, NO.7, p.1748-1752, (2015), ISSN:22295518

[37] X. Yu, X. Xie, *On Derivations of Black-Scholes Greek Letters*, Research Journal of Finance and Accounting, Vol.4,No.6, p.80-85, (2013), ISSN:22222847

Appendix

A MATLAB Code for GBM Stock Price Simulation

```
%Parameters
S0 = 1;          %start state/initial stock price
mu = 0.1;        %return
sigma = 0.05;    %constant volatility

nTrials = 100; %number of simulations
nPeriods = 500; %partition of T


%GBM
Geo = gbm(mu, sigma, "StartState",S0, "StartTime",0)

%Simulations and Plot
for i=1:nTrials

[X,T] = Geo.simulate(nPeriods, 'DeltaTime',1/nPeriods);
```

```
plot(T, X(:,:,1));
hold  on

end

xlim([0,1])

title("100 Stock Prices by Simulation of Geometric Brownian Motion")
xlabel("0 \leq t \leq 1")
ylabel("Stock Price")
```

B MATLAB Code for Heston Model Stock Price Simulation

```
%Parameters
S0 = 1;          %start state/initial stock price
mu = 0.1;       %return
sigma = 0.05;    %instantaneous volatility used in CIR stochastic volatility process
speed = 0.2;     %mean reversion speed (lambda in our Heston definition)
level = 0.1;     %mean reversion level (theta in our Heston definition)

nTrials = 100; %number of simulations
nPeriods = 500; %partition of T


%GBM
H = heston(mu,speed, level, sigma, "StartState",S0, "StartTime",0)

%Simulations and Plot
for i=1:nTrials

[X,T] = H.simulate(nPeriods, 'DeltaTime',1/nPeriods);


plot(T, X(:,1));
hold  on

end

xlim([0,1])
ylim([0,inf])
title("100 Simulations of Heston Model Stock Prices")
xlabel("0 \leq t \leq 1")
ylabel("Stock Price")
```

C MATLAB Code for Rough Heston Model Stock Price Simulation

```
%Parameters
S0 = 1;          %start state/initial stock price
mu = 0.1;        %return
sigma = 0.05;    %instantaneous volatility
speed = 0.2;     %mean reversion speed
level = 0.1;     %mean reversion level
alpha = -0.03;   %roughness index (-0.5, 0)

nTrials = 100; %number of MC simulations
nPeriods = 500; %partition of T


%GBM
RH = roughheston(mu,speed, level, sigma, alpha, "StartState",S0, "StartTime",0)

%Simulations and Plot
for i=1:nTrials

[X,T] = RH.simulate(nPeriods, 'DeltaTime',1/nPeriods);


plot(T, X(:,1));
hold  on

end

xlim([0,1])
ylim([0,inf])
title("100 Simulations of Rough Heston Model Stock Prices")
xlabel("0 \leq t \leq 1")
ylabel("Stock Price")
```

D Python Code for Black Scholes Implied Volatility Smile

```
import pandas as pd
import numpy as np
from scipy.stats import norm
import datetime as dt
from datetime import timedelta, datetime
import matplotlib.pyplot as plt
N = norm.cdf
M = norm.pdf


#Create Data Frames
```

```
OptionsDX_dataset = pd.read_excel("OptionsDX_pulled.xlsx", usecols="A,B,C,F,K,M,N,O,I")
OptionsDX_dataframe = pd.DataFrame(OptionsDX_dataset)
print(OptionsDX_dataframe)


#Clean Data Frames of na values
OptionsDX_dataframe.dropna()
OptionsDX_dataframe['OptionLength_Days'] = OptionsDX_dataframe[' [EXPIRE_DATE]']- \
OptionsDX_dataframe[' [QUOTE_DATE]']
print(OptionsDX_dataframe['OptionLength_Days'])

#T = 107/365
T = OptionsDX_dataframe['OptionLength_Days']
T=T.dt.days/365
r=0.0401
sigma = 0.3
print(T)


#Create volnew and volold to pass into Newton Raphson Loop
OptionsDX_dataframe['vol_new'] =  OptionsDX_dataframe[' [UNDERLYING_LAST]'] - \
OptionsDX_dataframe[' [UNDERLYING_LAST]'] + sigma
OptionsDX_dataframe['vol_old'] = OptionsDX_dataframe[' [UNDERLYING_LAST]'] - \
OptionsDX_dataframe[' [UNDERLYING_LAST]']


#implied vol Newton Raphson
Tol = 0.0000000001
iteration = 1000
print(OptionsDX_dataframe['vol_old'])
print(OptionsDX_dataframe['vol_new'])
for x in range(iteration):
    if (sum(abs(OptionsDX_dataframe['vol_new'] - OptionsDX_dataframe['vol_old'])) \
        < Tol): break
    OptionsDX_dataframe['vol_old'] = OptionsDX_dataframe['vol_new']
    OptionsDX_dataframe['d1'] = (np.log(OptionsDX_dataframe[' [UNDERLYING_LAST]']/ \
    OptionsDX_dataframe[' [STRIKE]']) + (r + OptionsDX_dataframe['vol_old']* \
    OptionsDX_dataframe['vol_old']/2)*T) / (OptionsDX_dataframe['vol_old']*np.sqrt(T))
    OptionsDX_dataframe['d2'] = OptionsDX_dataframe['d1']-  \
    OptionsDX_dataframe['vol_old'] * np.sqrt(T)
    OptionsDX_dataframe['BS_CALL'] = OptionsDX_dataframe[' [UNDERLYING_LAST]'] * \
    N(OptionsDX_dataframe['d1']) - OptionsDX_dataframe[' [STRIKE]'] * np.exp(-r*T)* \
    N(OptionsDX_dataframe['d2'])
    OptionsDX_dataframe['C'] = OptionsDX_dataframe['BS_CALL'] - \
    OptionsDX_dataframe[' [C_ASK]']
    OptionsDX_dataframe[' [C_VEGA]'] = (OptionsDX_dataframe[' [UNDERLYING_LAST]'] * \
    np.sqrt(T) * M(OptionsDX_dataframe['d1']))
    OptionsDX_dataframe['vol_new'] = OptionsDX_dataframe['vol_old']- \
    OptionsDX_dataframe['C']/OptionsDX_dataframe[' [C_VEGA]']
```

```
    print(x)
    print(sum(abs(OptionsDX_dataframe['vol_new'] - OptionsDX_dataframe['vol_old'])))
    print(OptionsDX_dataframe['vol_new'])
else: print("finally finished")

OptionsDX_dataframe['vol_new'].to_csv('output6.csv')
plt.scatter(OptionsDX_dataframe[' [STRIKE]'],OptionsDX_dataframe['vol_new'])
plt.xlabel("Strike Price (USD)")
plt.ylabel("implied vol")
plt.title("SPX B-S Pricing; S = 3952.01, r = 0.0401, Quote Date = 2023-03-01, \
Expire Date = 2023-06-16")
plt.show
```

E Python Code for Black Scholes Model

```
import pandas as pd
import datetime as dt
from datetime import timedelta, datetime
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
N = norm.cdf

OptionsDX_dataset = pd.read_excel("OptionsDX_pulled.xlsx", usecols="A,B,C,F,K,M,N,O,I")
OptionsDX_dataframe = pd.DataFrame(OptionsDX_dataset)
print(OptionsDX_dataframe)

OptionsDX_dataframe.dropna()
OptionsDX_dataframe['OptionLength_Days'] = OptionsDX_dataframe[' [EXPIRE_DATE]']-\
OptionsDX_dataframe[' [QUOTE_DATE]']
print(OptionsDX_dataframe['OptionLength_Days'])

T = OptionsDX_dataframe['OptionLength_Days']
T=T.dt.days/365
r=0.01
sigma = 0.30

#BS Call
OptionsDX_dataframe['d1'] = (np.log\
(OptionsDX_dataframe['[UNDERLYING_LAST]']/OptionsDX_dataframe[' [STRIKE]']) + (r \
+ sigma**2/2)*T) / (sigma*np.sqrt(T))
OptionsDX_dataframe['d2'] = OptionsDX_dataframe['d1'] - sigma * np.sqrt(T)
OptionsDX_dataframe['BS_CALL'] = OptionsDX_dataframe[' [UNDERLYING_LAST]'] * \
N(OptionsDX_dataframe['d1']) - OptionsDX_dataframe[' [STRIKE]'] * np.exp(-r*T)* \
N(OptionsDX_dataframe['d2'])

#BS Put
OptionsDX_dataframe['BS_Put'] = OptionsDX_dataframe[' [STRIKE]']*np.exp(-r*T)*\
```

```
N(-OptionsDX_dataframe['d2'])-OptionsDX_dataframe[' [UNDERLYING_LAST]']*\
N(-OptionsDX_dataframe['d1'])

#SPX B-S Scatter Plot
fig, ax1=plt.subplots()
ax2=ax1.twinx()
ax1.scatter(OptionsDX_dataframe[' [STRIKE]'],OptionsDX_dataframe['BS_CALL'], c='blue')
ax2.scatter(OptionsDX_dataframe[' [STRIKE]'], OptionsDX_dataframe['BS_Put'], c='red')
ax1.set_xlabel("Strike Price (USD)")
ax1.set_ylabel("B-S Call Price (USD)", c='blue')
ax2.set_ylabel("B-S Put Price (USD)", c='red')
plt.title("SPX B-S Pricing")
```