On Practitioner Implementation of Variance Gamma Option Pricing with Subprime Crisis Empirics

Honors Final Year Project

Course Code: MTH-490

Date of submission: 17 September, 2023

Department of Mathematics

University of Dhaka

Submitted by

Class Roll : AE-019-170

Registration No. : 2017-313-814

Session : 2017-18

Abstract

This project aims to show a new practitioner's option pricing model based on the Variance Gamma process. We conduct a comprehensive comparative analysis, evaluating this newly proposed model alongside the benchmark Black & Scholes (BS) model, the complicated Variance Gamma (VG) model, and the Practitioner's Black & Scholes (PBS) model applying on several thousand options over the periods of global financial crisis (subprime crisis) in 2008-2009. Our investigation encompasses an exploration of the significance, merits, and drawbacks associated with these four models. We utilize MATLAB's computational capabilities, leveraging the 'fmincon' and 'lsqnonlin' functions parallelly for calibrating the model parameters. This enables us to compute option prices, carry out RMSE comparisons with market prices, and evaluate model performance across diverse options, trading days, and moneyness levels. Our investigation consistently highlights the superior efficacy of the PVG model, particularly evident in its superior performance within in-sample data and its competitive standing within out-sample data.

Keyword: Black & Scholes, Variance Gamma, Practitioner, fmincon, lsqnonlin, moneyness levels, etc.

INDEX

Introduction

Option pricing model plays a vital role in finance by providing a structured approach to assessing the value of financial derivatives, particularly options. Efficient pricing and forecasting of call options in financial markets are of paramount importance to investors, practitioners, and academics alike. By allowing participants in the financial market to manage risk and maximize investment strategies, these derivative instruments are essential to modern finance.

The concept of the option pricing model can be traced back to the early 20th century when financial markets were evolving rapidly. In the scientific world, the share market is defined by Stochastic Price Movements and volatility variation, it is still difficult to compute option prices with any degree of accuracy. As a potent instrument for modeling asset returns and option pricing, the Variance Gamma (VG) model attained huge popularity as an alternative to the benchmark Black Scholes (BS) model.

In the early 1970s the groundbreaking Black-Scholes (BS) model was introduced. Developed by economists Fischer Black, Myron Scholes, and Robert Merton, this model revolutionized the financial industry by providing a mathematical framework to determine the fair market value of European-style derivatives.

The BS model serves as a foundational framework for option pricing, whereas the VG model introduces greater complexity, adapting to the dynamic financial market to enhance the value and relevance of option prices. It was introduced by Madan and Seneta in 1990 as a way to address some of the limitations of the Black-Scholes model, which assumes constant volatility and normally distributed returns. While the BS model operates as a diffusion model, the VG model represents a pure jump model lacking of diffusion components, marking a significant departure in modeling methodology.

The Practitioner formula is frequently preferred by financial professionals who place a high value on models that are reliable, adaptable, and capable of effectively portraying the practicality of financial markets. It is highly regarded for its capacity to produce accurate forecasts and useful insights compared to conventional models like BS and VG. Drawing inspiration from the Practitioner BS model, we conduct a thorough empirical analysis of several thousand historical market data spanning the 2009 subprime crisis to assess the PVG model's effectiveness in predicting option prices making comparisons with BS, VG, and PVG models.

This project primarily aims to explore the PVG sigma model and subsequently evaluate it in comparison to benchmark models such as BS, it's alternative VG, and PBS. We delve into each model's methodology, calibration, and RMSE calculation to determine their respective performance.

1. The Black-Scholes Model

1.1 Description

The Black-Scholes model determines a stock's theoretical price in options trading. It is used for both call and put options. The model relies on five variables for price calculation: underlying asset's price, strike price, risk-free rate, volatility, and expiration time. It is only applicable to European options trading.

The Black-Scholes theory was developed by economists Fischer Black and Myron Scholes in 1973. It is the most common options trading model and binomial model. The model is based on many assumptions limiting its usage outside European options trading.

- Black-Scholes is a pricing model used in options trading. It derives the fair price of a stock.
- Fischer Black and Myron Scholes met at the Massachusetts Institute of Technology (MIT). Their pricing model completely revolutionized technical investing. Black and Scholes won the Nobel Prize for their contribution in 1997.
- Black and Scholes assume that there are no market arbitrage opportunities or riskless profits. In real-world scenarios, volatility is not constant across time; transaction costs exist.
- Real-world data depicts that price returns tend to have a skewed distribution; prices fall much faster than they rise.

It is used in real life in many contexts. It is used for accounting and regulation, typically with some kind of realized or estimated volatility substituted for implied volatility.

1.1.1 Black-Scholes Model for Option Pricing

The Black-Scholes Model was developed by economists Fischer Black and Myron Scholes in 1973. The Black-Scholes model works on five input variables: underlying asset price, strike price, risk-free rate, volatility, and expiration time.

It is a mathematical model that utilizes a partial differential equation to calculate the price of options. This partial differential is known as the Black-Scholes equation. Banks and Financial institutions use this model for evaluating European options. The primary objective behind the model is to hedge options in a portfolio and eliminate the risk factor.

Fischer Black and Myron Scholes met at the Massachusetts Institute of Technology (MIT) and started a partnership that lasted 25 years. Their pricing model completely revolutionized technical investing. Black and Scholes won the Nobel Prize for their contribution in 1997.

Black and Scholes assume there are no market arbitrage opportunities or riskless profits. This is why the model receives criticism. In real-world scenarios, volatility is not constant across time; transaction costs exist. Real-world data depicts that price returns tend to have a skewed distribution; prices fall much faster than they rise.

At the beginning of the 20th century, French mathematician Louis Bachelier made an analogy between Brownian motion and the movement of financial assets in his Theory of Speculation. The Black-Scholes theory incorporates this assumption.

1.1.2 Black-Scholes Assumptions

Black-Scholes model assumptions are as follows

- Black-Scholes theory assumes that option prices exhibit Brownian motion.
- The model assumes that risk-free rates are constant. In reality, they are dynamic—they fluctuate with supply and demand
- The theory assumes stock returns resemble a log-normal distribution
- It also assumes that we have a frictionless market; and that there are no transaction costs, which is not the case with real-world scenarios.
- Black and Scholes neglect dividend payouts throughout the option period

1.1.3 Advantage of the Black-Scholes model

Accuracy

The first and foremost advantage of the Black-Scholes model is that it is one of the most accurate option pricing models which is available as compared to other option pricing models. It takes into account all of the factors that can affect the price of an option, including time to expiration, volatility, interest rates, and other important variables which in turn leads to better or accurate pricing of options contracts and thus gives this model an edge over others when it comes to pricing of options.

Speed

Another benefit of the Black-Scholes model is that it can price the options contracts very quickly as it uses a mathematical formula to calculate the price of an option. This makes it a popular choice for traders as in the case of stock markets the prices change in seconds and traders need to make quick decisions about whether or not to buy or sell an option contract based on the price of the underlying which are stocks or index in case of stock markets.

Flexibility

Another important advantage of the Black-Scholes model is that its use is not limited to the stock market only rather it is flexible enough to be used in a variety of different situations. Hence one can use this model to price options not only on stocks but also bonds, commodities, and other assets which makes it a potent tool for traders who want to trade in a variety of assets rather than trading in only one market or asset class

1.1.4 Disadvantage of the Black Scholes Model

Ignores Other Factors

The biggest disadvantage of the Black-Scholes model is that while calculating the theoretical value of the option contract it only takes into account the price of the underlying asset and ignores the other important factors. Hence for example it does not take into account other factors like dividends, interest rates, changes in volatility, external shocks, market regulator actions, and so on. This in turn can lead to some inaccuracies in the pricing of options contracts thus defeating the whole idea of using this model for fair pricing of options contracts.

1.2 Methodology

1.2.1 Black - Scholes Merton formula

Black & Scholes model is the most famous option pricing model for its simplicity, closed-form solution, and ease of implementation. It was introduced by Black & Scholes in 1973. This model works on European option prices where interest rates and volatility are constant. The fundamental method of the Black & Scholes model is based on geometric Brownian motion which is completely free from imperfections. We need some parameters, which are

S= stock price
K= strike price
σ = volatility
T = time of maturity
R = risk-free interest rate

Therefore, European call option can be written as a function of S & t

$$C(S_t,t)= \mathbb{E}[e^{-r(T-t)}(S_T - K)|\mathcal{F}_t]$$

The equation

$$S_t = S_0 e^{\int_0^t 6 dW + \int_0^t (r - \frac{6^2}{2}) ds}$$

can be written as

$$S_T = S_0 e^{\int_t^T бdW + \int_t^T (r-\frac{б^2}{2})ds}$$
$$= S_0 e^{б(W_T - W_t) + \left(r-\frac{б^2}{2}\right)(T-t)}$$
$$= S_0 e^{-б\sqrt{\tau}Y + \left(r-\frac{б^2}{2}\right)\tau}$$

Letting, (T-t) = $\boldsymbol{\tau}$ and define Y=$-\frac{W_T - W_t}{\sqrt{T-t}}$

Since S_t is measurable of $\mathcal{F}_t$ and $e^{-б\sqrt{\tau}Y + \left(r-\frac{б^2}{2}\right)\tau}$ is independent of $\mathcal{F}_t$, then we can write,

$$C(t, s) = \mathbb{E}[e^{-r(T-t)}(S_T - K)]$$
$$= \frac{1}{\sqrt{2л}} \int_{-\infty}^{\infty} e^{-r\tau}(se^{-б\sqrt{\tau}Y + \left(r-\frac{б^2}{2}\right)\tau} - K)\, e^{-\frac{y^2}{2}} dy \text{ .... (1)}$$

Since,

$$e^{-б\sqrt{\tau}Y + \left(r-\frac{б^2}{2}\right)\tau} > \frac{K}{s}$$

or, $-б\sqrt{\tau}Y + \left(r - \frac{б^2}{2}\right)\tau > \ln\frac{K}{s}$

or, $-б\sqrt{\tau}Y > -\left(r - \frac{б^2}{2}\right)\tau + \ln\frac{K}{s}$

or, $б\sqrt{\tau}Y < \ln\frac{s}{K} + \left(r - \frac{б^2}{2}\right)\tau$

or, $Y < \frac{1}{б\sqrt{\tau}}(\ln\frac{s}{K} + \left(r - \frac{б^2}{2}\right)\tau)$

or, $Y < d_2$

letting, $d_2 = \frac{1}{б\sqrt{\tau}}(\ln\frac{s}{K} + \left(r - \frac{б^2}{2}\right)\tau)$

and $d_1 = d_2 + б\sqrt{t}$

$$= \frac{1}{б\sqrt{\tau}}(\ln\frac{s}{K} + \left(r - \frac{б^2}{2}\right)\tau) + б\sqrt{t}$$

$$= \frac{\ln\frac{s}{K} + \left(r - \frac{б^2}{2}\right)\tau + б^2 t}{б\sqrt{t}}$$

$$= \frac{1}{б\sqrt{\tau}}(\ln\frac{s}{K} + \left(r + \frac{б^2}{2}\right)\tau)$$

Now, from (1)

$$C(s,t) = \frac{1}{\sqrt{2л}} \int_{-\infty}^{\infty} e^{-r\tau}(Se^{-б\sqrt{\tau}y + \left(r-\frac{б^2}{2}\right)\tau} - K)\, e^{-\frac{y^2}{2}} dy$$

$$= \frac{1}{\sqrt{2л}} \int_{-\infty}^{d_2} Se^{-б\sqrt{\tau}y - \frac{б^2}{2}\tau - \frac{y^2}{2}} dy - \int_{-\infty}^{d_2} K\, e^{-(r\tau + \frac{y^2}{2})} dy$$

$$= \frac{1}{\sqrt{2\pi}}\left[\int_{-\infty}^{d_2} Sdy - \int_{-\infty}^{d_2} K\, e^{-(r\tau+\frac{y^2}{2})}dy\ \right]$$

$$= \frac{S}{\sqrt{2\pi}}\left[\int_{-\infty}^{d_2+\sigma\sqrt{\tau}} e^{-\frac{1}{2}z^2}dz - \frac{Ke^{-r\tau}}{\sqrt{2\pi}}\int_{-\infty}^{d_2} e^{-\frac{y^2}{2}}\, dy\ \ [\text{assume}, \sigma\sqrt{\tau} + y = z]\right.$$

$$= SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

Put call parity:

The relationship between the prices put & call option having the formula

$$S_t + P_t - C_t = Ee^{-r(T-t)}$$

1.2.2 Derivation Black & Scholes Model

Let,

V=option valuation.
P=put option
C=call option

from GBM, we know that,

$dS=\mu Sdt + \sigma SdB$.........(1)

and

$$dv = \frac{dv}{dt}dt + \frac{dv}{ds}ds + \frac{1}{2}\frac{d^2v}{ds^2}ds^2$$

$$= \frac{dv}{dt}dt + \frac{dv}{ds}(\mu Sdt + \sigma SdB) + \frac{1}{2}\frac{d^2v}{ds^2}\sigma^2 s^2 dt \quad [\text{ substitute (1) }]$$

$$= (\frac{dv}{dt} + \mu S\frac{dv}{ds} + \frac{1}{2}\frac{d^2v}{ds^2}\sigma^2 s^2)dt + \sigma SdB \text{ ....(2)}$$

Value of the portfolio that is long,

$P = \Pi S - V$...........(3)

or, $dP = \Pi dS - dV$

$$= \Pi(\mu Sdt + \sigma SdB)\text{-}(\frac{dv}{dt} + \mu S\frac{dv}{ds} + \frac{1}{2}\frac{d^2v}{ds^2}\sigma^2 s^2)dt + \sigma SdB]$$

$$=(\Pi\mu S - \frac{dv}{dt} - \mu S\frac{dv}{ds} - \frac{1}{2}\frac{d^2v}{ds^2}\sigma^2 s^2)dt + (\Pi\sigma S - \sigma S\frac{dv}{ds})dB$$

Letting

$$\Pi\sigma S - \sigma S\frac{dv}{ds}=0$$

or, $\Pi - \frac{dv}{ds}=0$

or, $\Pi = \frac{dv}{ds}$...........(4)

Therefore,

$$dP=(\Pi\mu S - \frac{dv}{dt} - \mu S\frac{dv}{ds} - \frac{1}{2}\frac{d^2v}{ds^2}\sigma^2 s^2)dt$$

Since the portfolio is riskless then

$$dP = Prdt$$

or, $\Pi\mu S - \frac{dv}{dt} - \mu S\frac{dv}{ds} - \frac{1}{2}\frac{d^2v}{ds^2}6^2s^2)dt = Prdt$

or, $(\Pi\mu S - \frac{dv}{dt} - \mu S\frac{dv}{ds} - \frac{1}{2}\frac{d^2v}{ds^2}6^2s^2) = Pr$

or, $\frac{1}{2}\frac{d^2v}{ds^2}6^2s^2 + \frac{dv}{dt} + rS\frac{dv}{ds} = rv$ [using (3) & (4)]

2. Practitioner Black & Scholes (PBS) Model

The model is referred to as the Practitioner Black-Scholes (PBS) model by Christof-Fersen and Jacobs and as the ad-hoc model by Dumas, Fleming and Whaley. It constitutes a simple way to price options, based on implied volatilities and the Black-Scholes pricing formula. The assumption of constant volatility in the Black-Scholes model is circumvented by using volatility that is not constant, but rather depends on moneyness and maturity.

Implementation of Practitioner Black and Scholes Model:

There are a number of possible ways of implementing Practitioner Black Scholes method. One, which we shall call "standard PBS", is a two-stage approach:

1) Use the cross-section of options to estimate a linear regression model with implied volatility as the dependent variable, and strike price, time to expiry, and their squares and cross-product, as explanatory variables.
2) Plug the fitted values from this regression into the Black-Scholes formula to obtain predicted option prices.

A problem we identify with this method is one of model incoherency: the IV equation that is estimated does not correspond to the set of option prices used to estimate it. We use the Monte Carlo method to verify that

1) Standard PBS gives biased option values, both in-sample and out-of-sample;
2) Using standard (log-linear) PBS with smearing almost completely eliminates the bias
3) NLLS gives biased option values, but the bias is less severe than with standard PBS.

We are led to conclude that, of the range of possible approaches to implementing PBS, log-linear PBS with smearing is preferred on the basis that it is the only approach that results in valuations with negligible bias.

2.2 Methodology

Firstly, we estimate the implied volatility for each option using Black & Scholes formula. The following equation is

$C_i = C_i^{BS}(6_i)$ ....(1)

Where,

C_i=market price of call option i

C_i^{BS}=Black Scholes valuation of call option i^2.

Secondly, applying simple order least square (OLS) to predict the implied volatility with time maturity and strike price on different polynomial which is also known as DFW (Dumas, Fleming, Whaley) implied volatility equation. They consider as Deterministic Volatility Function (DVF) which is a parabolic shape of volatility smile.

$$\sigma(\theta)=\theta_0+\theta_1 K+\theta_2 K^2+\theta_3 T+\theta_4 T^2+\theta_5 KT+\varepsilon_{IV} \text{..........(2)}$$

σ =Implied volatility

K= strike price

T = time maturity

$\theta_0, \theta_1, \theta_2, \ldots\ldots$ = model parameter

Finally, obtaining the fitted value of the implied volatility is

$$\sigma(\hat{\theta})=\hat{\theta}_0+\hat{\theta}_1 K+\hat{\theta}_2 K^2+\hat{\theta}_3 T+\hat{\theta}_4 T^2+\hat{\theta}_5 KT \text{...........(3)}$$

where caps indicate the OLS estimates. The above formula can be written as PBS format

$$C^{PBS}=C^{BS}((\hat{\theta})) \text{ ....(4)}$$

Then the log linear form of the implied volatility is

$$\ln[\sigma(\theta)] = \theta_0+\theta_1 K+\theta_2 K^2+\theta_3 \tau+\theta_4 \tau^2+\theta_5 K\tau+\varepsilon_{IV} \text{.............(5)}$$

using (4) & (5), we get a new equation,

$$C^{PBS,\log IV} = C^{BS}[e^{\ln(\sigma(\hat{\theta}_{\log IV}))}] \text{.........(6)}$$

An another alternative method is Non Linear Least square (NLLS) which helps to express PBS method by using market option pricing &estimate the parameter. We can easily understand this method from Monte-Carlo method. This method emphasizes standard PBS that is log linear because it gives unbiased valuation both in sample & out sample and performs well.

Consider the regression model

$$y_i = x_i b + u_i, i = 1,2, \ldots n \;; \ldots \ldots \ldots (7)$$

where $\hat{u}_i$ is residual.

We determine f(y) which is non-linear function but it gives a biased estimation until it can be linear. The smearing formula is proposed by Duan is

$$\hat{f}_i = \frac{1}{n}\sum_{j=1}^{n} f(\hat{y}_i + \hat{u}_j) \text{.........(8)}$$

Expressing it PBS model using equation (4) as similar type,

$C_i^{PBS,BS} = \frac{1}{n}\sum_{j=1}^{n} C_i^{BS}(6_i(\widehat{y_\imath}) + \widehat{\varepsilon_\imath})$.........(9)

Then

$C_{n+1}^{PBS,BS} = \frac{1}{n}\sum_{j=1}^{n} e^{\ln((6_i(\widehat{\Theta_\imath})+\widehat{\varepsilon_j}))}$.........(10)

Now we extend the smearing formula n+1 term to get

$$C_{n+1}^{PBS,\log IV,SM} = \frac{1}{n+1}\sum_{i=1}^{n+1} C_i^{BS} e^{\ln((6_{n+1}(\widehat{\Theta_{\log IV}})+\widehat{\varepsilon_j}))}\text{.........(11)}$$

the out-sample option $\widehat{\varepsilon_{n+1}}$=0.

Now, we evaluate option pricing data using NLLS in the implied volatility equation.
NLLS applying in equation (2),

$\widehat{\Theta_{NLLS}}$= argmin $\sum[C_i - C_i^{BS}(6_i(\Theta))]^2$

the problem can be expressed as

$$C_i = C_i^{BS}\big(6_i(\Theta)\big) + \varepsilon_i$$

removing error term or suppose $\varepsilon_i = 0$,we get,

$$C_i = C_i^{BS}\big(6_i(\Theta)\big)$$

but the above formula cannot give a properly correct solution. If we apply the Hausman test, it gives an approximately unbiased answer. The Hausman test is

$$H= \frac{1}{(\hat{V}_{\log IV} - \hat{V}_{NLLS})}(\hat{\theta}_{\log IV} - \hat{\theta}_{NLLS})(\hat{\theta}_{\log IV} - \hat{\theta}_{NLLS})'$$

Where $\hat{V}_{\log IV}$&$\hat{V}_{NLLS}$ are estimator of the estimated variance matrices.

3. The Variance Gamma Process

3.1 Description

The Variance Gamma model is a flexible model that can be used to price options in a variety of market condition. It was introduced as an extension of Geometric Brownian Motion to overcome some issues that the Black and Scholes model has in pricing options. It has been known in the financial literature for several years.

The first complete presentation of this model is due to Madan and Seneta in 1990. The model presented in this paper is however a symmetric variance gamma model. In 1991 Madan and Milne published a paper, where they study equilibrium option pricing for the symmetric variance gamma process using a representative agent model under a constant relative risk aversion utility function. This general non symmetric process is described more completely in the 1998 paper by Madan, Carr and Chang where also a closed form solution for European vanilla options is presented.

Variance Gamma process is an example of Lévy process. It can be defined as the exponential of a Lévy process with normally distributed increments. Let's break down the components of this definition:

1. Lévy process: A Lévy process is a stochastic process that generalizes Brownian motion by allowing for jumps in addition to continuous paths. In a Lévy process, the increments over non-overlapping time intervals are statistically independent and have a certain distribution, known as the Lévy distribution. This distribution characterizes the jump component of the process. The jumps can have various sizes and occur at random times, leading to discontinuities in the paths of the process.

2. Normally distributed increments: The increments of the Lévy process are assumed to follow a normal distribution, which makes the VG process especially appealing since it allows the incorporation of familiar statistical tools and methods from the normal distribution.

3. Exponential function: By taking the exponential of the Lévy process, we ensure that the VG process has strictly positive values. This property is crucial when modeling financial asset prices, as they cannot take negative values.

4. Asset price: asset price refers to the value of a financial instrument, such as a stock, bond, or option that is being modeled. It is assumed to follow a stochastic process which incorporates random fluctuations, volatility, and other statistical properties that are observed in real-world financial markets.

3.2 Methodology

The VG process is often used to model stock price movements, exchange rates, or other financial variables with similar characteristics.

The model presents two additional parameters, compared with the Geometric Brownian Motion, which allows for control of the skewness and the kurtosis of the distribution of stock price returns. We can define a variance gamma process, $X(t; \sigma, \nu, \vartheta)$ as a Brownian motion where the time is stochastic and it is given by a gamma process with unit mean rate, $\gamma(t; 1, \nu)$, in compact form we can write:

$$X(t;\ \sigma, \nu, \vartheta) = b(\gamma(t;\ 1, \nu);\ \vartheta, \sigma$$

The three parameters involved in the VG model are:

σ: volatility of Brownian motion which control volatility
υ: variance rate of gamma time changes which controls kurtosis

θ: the drift in Brownian motion which control skewness

The density function for variance gamma process at a time t can be expressed as a normal density function conditional on the realization of the time change given by the gamma distribution. However the unconditional density function for variance gamma can be written as:

$$f_{X(t)}(X) = \int_0^{+\infty} \frac{1}{\sigma\sqrt{(2\pi g)}} \exp\left(-\frac{(X-\vartheta g)^2}{2\sigma^2 g}\right) \frac{g^{\frac{t}{v}-1}\exp\left(-\frac{g}{v}\right)}{v^{\frac{t}{v}}\Gamma\left(\frac{t}{v}\right)} dg$$

Here $g = \text{time increment}$;

In the same way, the characteristic function for the variance gamma process can be expressed conditional on the gamma time.

The unconditional characteristic function, $\varphi X(t)(u) = E\left[\exp\left(iuX(t)\right)\right]$, is:

$$\phi_{X(t)}(u) = \left(\frac{1}{1 - i\vartheta vu + \left(\frac{\sigma^2 v}{2}\right)u^2}\right)^{\frac{t}{v}}$$

The statistical stock price dynamic can be obtained by replacing the geometric Brownian motion with the variance gamma process in the equation used by Black and Scholes in their famous model. We have then the following process:

$$S(t) = S(0)\exp\left[mt + X\left(t;\sigma_S, v_S, \vartheta_S\right) + \omega_S t\right]$$

where $X(\cdot)$ is a variance gamma process, m is the mean rate of return on the stock under the statistical probability measure and the subscripts "S" are used to stress the fact that the parameters are the statistical ones. The values of ω is determined as a non arbitrage condition, by evaluating the characteristic function for $X(t)$ at $u = 1/i$ so that

$$E[S(t)] = S(0)\exp(mt) \Leftrightarrow E\left[\exp(X(t))\right] = \exp(-\omega St)$$

and it is equal to

$$\omega_S = \frac{1}{v_S}\ln\left(1 - v_S\vartheta_S - \frac{\sigma^2 v_S}{2}\right)$$

As we know, the price of a European call options $C(S(0), K, t)$, where K is the strike and t is the maturity, can be written with the familiar expression:

$$C(S(0), K, t) = e^{-rt}E\left[\max(S(t) - K, 0)\right]$$

Where the expectation is taken under the risk neutral process.

The variance gamma can be used to price European call options. It is not moneyness biased, however it still presents problems related with the estimation of options with different maturities.

4. Practitioner Variance Gamma (PVG) Process
4.1 Description

The practitioner's variance gamma process refers to the application and utilization of the VG process by professionals in the quantitative finance field. It is a modified and simplified version of the original Variance Gamma process that was developed to address certain limitations and practical considerations encountered in real-world financial modeling. The Practitioner's VG process introduces adjustments to the original VG process to make it more suitable for practical applications and align it better with empirical observations in financial markets. Offering a flexible framework for modelling asset prices, pricing derivatives accurately, and managing financial risks Practitioner's VG process improves the modeling of jump behavior and incorporates more realistic features observed in asset price dynamics.

Here are some key characteristics and modifications associated with the Practitioner's VG process:

1. Fine-Tuning of Parameters: The Practitioner's VG process allows practitioners to adjust the parameters of the VG process to better match the observed market behavior. This flexibility enables them to calibrate the model to specific assets or market conditions and capture the empirical features more accurately.

2. Skewness Adjustment: The original VG process assumes symmetric jumps, meaning that the jump sizes have equal probabilities of being positive or negative. In contrast, the Practitioner's VG process allows for skewness adjustments, acknowledging the observed asymmetry in jump sizes. By introducing skewness, the model can better represent the real-world behavior of asset prices.

3. Tail Behavior: The original VG process assumes a Gaussian tail distribution for the jump sizes. However, empirical evidence suggests that financial asset returns often exhibit heavier tails than the Gaussian distribution. The Practitioner's VG process allows for modifications to the tail behavior, such as introducing fat-tailed distributions or incorporating power-law decay.

4. Higher Moment Adjustment: In addition to capturing skewness, the Practitioner's VG process enables adjustments to higher moments, such as excess kurtosis. This modification helps address the observed phenomenon of heavy-tailed and peaked distributions in asset returns, which are not adequately captured by the original VG process.

5. Calibration to Market Data: The Practitioner's VG process emphasizes the importance of calibrating the model to market data. By estimating the parameters from historical or implied market data, practitioners can ensure that the model aligns with the observed statistical properties and reproduces the market prices of derivatives accurately.

4.2 Classification of PVG

Overall, the Practitioner's VG process provides practitioners with a more flexible and customizable framework for modeling asset price dynamics. The parameters of the Variance

Gamma (VG) process, often used in the practitioner's framework, represent key elements that govern the behavior and characteristics of the model.

The three parameters associated with the jump component are (1) Practitioner VG sigma, (2) Practitioner VG theta, (3) Practitioner VG nu
We know that, Practitioner’s VG process allows a practitioner to adjust the parameters of the VG process. There are three parameters in the VG process: σ, θ and υ. Here, we will use implied volatility σ_p. Then we will have three sets of changes in the adjustment of the motion. They are:

1. Practitioner’s VG- sigma.
2. Practitioner’s VG- theta.
3. Practitioner’s VG- nu.

1) Practitioner’s VG (PVG)- σ :

σ is the volatility of the BM which controls the volatility of the VG process. But now we use the relation between Variance Gamma and implied volatility and σ will depend on σ_p and θ, υ. So now the implied volatility σ_p will control the volatility of the whole process

$$\sigma^2 = \sigma_p^2 - \theta\upsilon$$

$$\underline{\mu} + \underline{\theta} = \mu - \sigma_p^2/2$$

Thus, named Practitioner’s VG- sigma process.

2)Practitioner’s VG (PVG)-θ :

θ is the drift of the BM which controls the skewness of the VG process. We will now use the implied volatility σ_p to control the drift of the VG process by using the relation between Variance Gamma and implied volatility and θ will be dependent on σ_p and σ,υ. So σ_p will control the skewness of the VG process.

$$\theta = (\sigma_p^2 - \sigma^2)/\upsilon$$

$$\underline{\mu} + \underline{\theta} = \mu - \sigma_p^2/2$$

Thus, named Practitioner’s VG theta process.

3)Practitioner’s VG-μ:

υ is the drift of the BM which controls the kurtosis of the VG process. We will now use the implied volatility σ_p to control the drift of the VG process by using the relation between Variance Gamma and implied volatility and υ will be depended on σ_p and θ, σ. So σ_p will control the kurtosis of the VG process.

$$\upsilon = (\sigma_p^2 - \sigma^2)/\theta$$

$$\underline{\mu} + \underline{\theta} = \mu - \sigma_p^2/2$$

Thus, named Practitioner’s VG nu process.

Here, σ = Volatility of BM
θ= drift of the BM
υ= variance of time shift, which is defined by the subordinating Gamma process
σ_p= Implied Volatility
$\underline{\mu}$ = Mean of Black and Scholes
$\underline{\theta}$ =Drift of Black-Scholes (which is constant)
μ= Mean of Variance Gamma Process

5. Analysis

5.1 Dataset Introduction and Data Filtering

Option Price data is collected through various sources like stock market, online brokers, websites, financial institutions, etc. For this project, we've used a dataset that contains call option price data of Wednesdays and Thursdays of 2009. The dataset contains the following columns:

1. Stock Price - Price of the underlying stocks at time t
2. Exercise Price - Price at which underlying stock can be bought or sold
3. Risk-free Interest
4. Maturity – The date at which the option ends or must be exercised
5. Market Prices – Call option prices recorded

Here we've used Wednesday's data as In-Sample data and for Out-Sample data we've used Thursdays. There are 44,598 data available for Wednesday and 44,097 data for Thursday.

Data filtering is a process of selectively extracting or retaining specific subsets of data from a larger dataset based on certain criteria or conditions. It is a fundamental step in data analysis and is often performed to focus on relevant information for a particular analysis or model. Data collection and acquisition often introduce errors in data i.e. missing values, typos, mixed formats, replicate entries for the same real-world entity, outliers. So data cleaning is a must done by examining the dataset to exclude, rearrange or apportion data.

For our dataset cleaning, we followed three filtering conditions of Call Option pricing:

(1) Maturity should be 10 days or above;
(2) Moneyness is the relative position of the current price of an underlying asset with respect to the strike price. Here we took moneyness as 0.9 to 1.1 and 0.85 to 1.15;
(3) Strike and maturity options are considered as a similar option;

Following the dataset filtration process based on these conditions, we obtained the following No. of data for each moneyness level, which served as the basis for our calibration and subsequent analysis:

1. Moneyness 0.9 to 1.1:
   In-Sample: 7,739
   Out-Sample: 7,619

2. Moneyness .85 to 1.15:
   In-Sample: 10,759
   Out-Sample: 10,77

5.2 Analyzing Models and Error Evaluation

At first, we used In-Sample data to calibrate the parameters Sigma, Theta, Nu as required for respective models and the regression coefficients required to calculate Practitioners Implied Volatility. Then we calculated Option Prices using the four models BS, VG, PBS, and PVG. (See MATLAB codes in Appendix: Page - 47)

We used these calibrated parameters and regression coefficients to calculate the Practitioner Implied Volatility and Call Option Prices of Out-Sample data.

The parameters associated with each model and their respective calibrations are as follows:

1. Black & Scholes: Sigma
2. Variance Gamma: Sigma, Theta, Nu
3. Practitioners Black & Scholes: Regression Coefficients of Practitioners Implied Volatility
4. Practitioners Variance Gamma: Sigma, Theta, Nu, Regression Coefficients of Practitioners Implied Volatility

After that, we compared the predicted call options we got from the models and the recorded market prices. To evaluate the error between predicted and recorded call option prices we used RMSE (Root-mean-squared Error).

Table 5.2.1: Comparing RMSE values of Discussed Models

DATE	*BS*	*VG*	*PBS*	*PVG SIGMA*	*NO. OF DATA*
2-SEP	7.701727	1.991546	0.92718	0.921287	174
3-SEP	5.878816	2.616458	2.504922	2.517947	162

For instance, in a single-day calibration scenario, let's consider the data presented in Table 5.2.1. This table showcases the Root Mean Square Error (RMSE) values associated with various models when applied to in-sample data on September 2nd and out-sample data on September 3rd. After rigorous filtering, we were left with 174 data points for in-sample calibration, and the results indicated that the PVG Sigma model yielded the lowest RMSE, followed by PBS, VG, and BS models.

For the out-sample data on September 3rd, we had 162 data points to calibrate. Surprisingly, the PBS model demonstrated the best performance in this scenario, followed by PVG Sigma and the other models.

This process of calculating RMSE for each day was repeated across all the days in our dataset, and the findings were consolidated into a comprehensive results table in the dedicated results section of our project report.

The figure below contains graphs that show the comparison of Predicted Call Option prices using different models and Exact Market prices in both In-sample and Out-sample data where the X-axis indicates the index of option pricing parameters (Strike price, Maturity, Interest rate, Stock Prices) of that date after all the filtering.

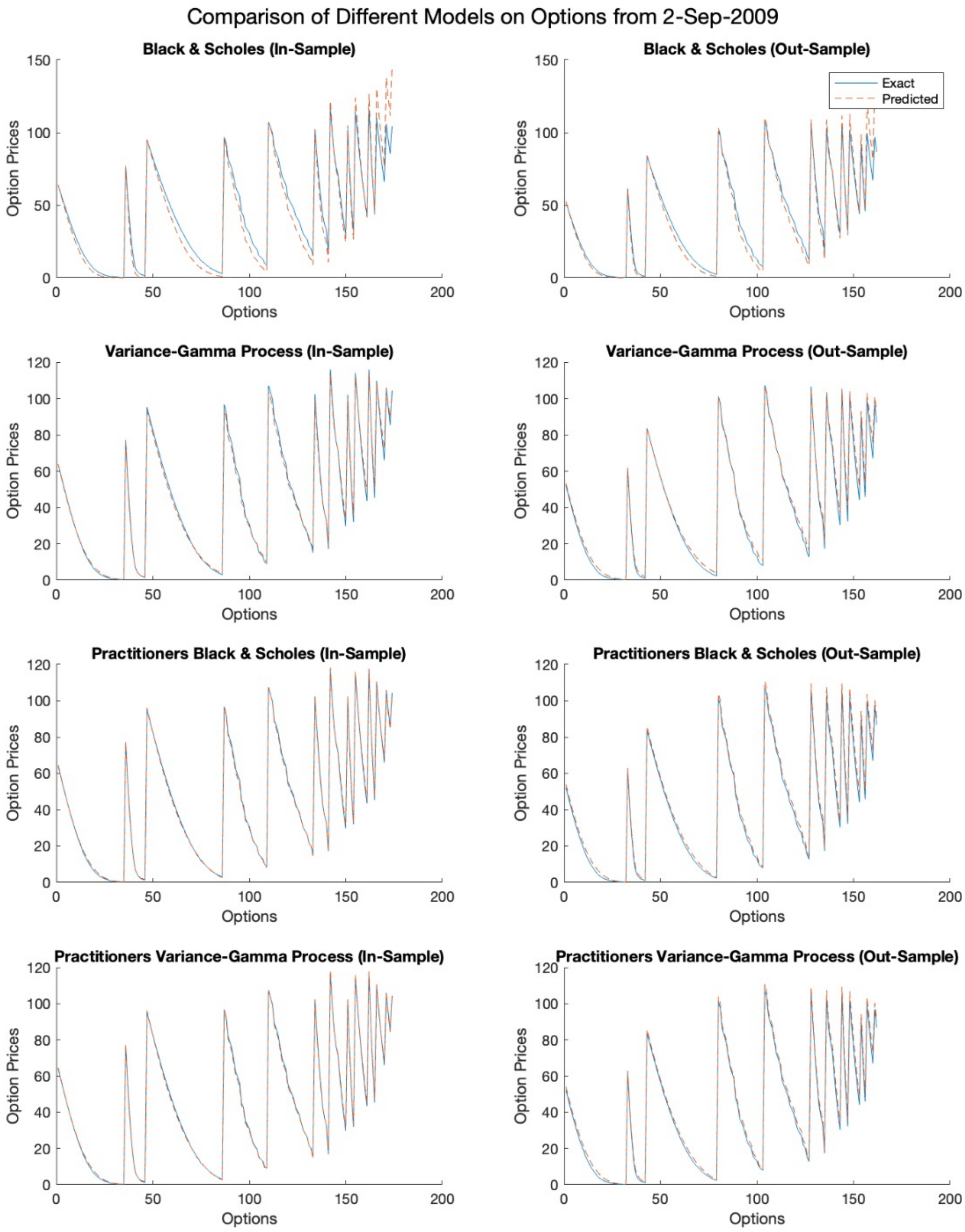

Figure 5.2.1: <u>Predicted Market Price vs Original Market Price on 2 September, 2009 for different models</u>

6. Results:

6.1 Model Evaluation Using 'fmincon'

6.1.1 Introduction of MATLAB function 'fmincon'

'fmincon' is a nonlinear programming solver that works on nonlinear constraints optimization and trying to determine constrained minimization which is the combination of scalar function with different variables. It is a gradient-based method.

The general equation of the optimization problem is

$$f(x) = \begin{cases} c(x) \leq 0, c_{eq} = 0 \\ Ax \leq 0, \quad A_{eq}x = 0 \\ l_b \leq x \leq u_b \end{cases}$$

For solving optimization problem , we need an objective function which are needed to calculate the minimum value of the function along with the following parameters:

$$\begin{aligned} &x_0 = \text{initial point of x which is scalar, vector, matrix} \\ &A_{ineq} = \text{matrix for linear inequality constraint} \\ &b_{ineq} = \text{vector for linear inequality constraint} \\ &A_{eq} = \text{matrix for linear equality constraint} \\ &b_{eq} = \text{vector for linear equality constraint} \\ &c_{eq} = \text{return vector for linear equality constraint} \\ &l_b = \text{vector of lower bound} \\ &u_b = \text{vector of upper bound} \end{aligned}$$

Formatting 'fmincon' & its description :

1) Our first attempt is to find the minimum value of f(x) by using x_0.
2) If no equality exist then A = [ ] and b = [ ].
3) If no inequality exist then A_{eq} = [ ] and b_{eq}= [ ].
4) If x(i) is unbounded below and above then respectively $l_b(i) = -\text{inf}$ and $u_b = \text{inf}$.
5) If no bounds exist then l_b= [ ] and u_b = [ ].
6) If no nonlinear inequality or equality constraints exist then nonlcon =[ ].

Equation Distribution	Description
Input Based Arguments	
1)$x = \text{fmincon}(f, x_0, A, b)$	minimizing x that discussed subject f to the linear inequalities $Ax \leq b$.
2)$x = \text{fmincon}(f, x_0, A, b, A_{eq}, b_{eq})$	minimizing the subject f with

	linear equalities $A_{eq}x = b_{eq}$ and $Ax \leq b$.
3)$x = \text{fmincon}(f, x_0, A, b, A_{eq}, b_{eq}, l_b, u_b)$	finding x when upper and lower bound defined such that $l_b \leq x \leq u_b$.
4)$x = \text{fmincon}(f, x_0, A, b, A_{eq}, b_{eq}, l_b, u_b, \text{nonlcon})$	minimizig nonlinear inequalities $c(x) \leq 0$ and equalities $c_{eq}(x) = 0$.
5)$x =$ $\text{fmincon}(f, x_0, A, b, A_{eq}, b_{eq}, l_b, u_b, \text{nonlcon}, \text{options})$	minimized optimization option
Output Based Arguments	
6)$[x, \text{fval}] = \text{fmincon}(\dots)$	giving the objective function f value at the solution x.
7)$[x, \text{fval}, \text{exitflag}] = \text{fmincon}(\dots)$	giving the value of exitflag which discuss the exit condition.
8)$[x, \text{fval}, \text{exitflag}, \text{output}] = \text{fmincon}(\dots)$	giving a structure output the instruction regarding as optimization
9)$[x, \text{fval}, \text{exitflag}, \text{output}, \text{lambda}] = \text{fmincon}(\dots)$	provides a formation lambda which contains Lagrange multiplier at the solution x.
10)$[x, \text{fval}, \text{exitflag}, \text{output}, \text{lambda}, \text{gradient}] =$ $\text{fmincon}(\dots)$	provides gradient value of f at the solution x
11)$[x, \text{fval}, \text{exitflag}, \text{output}, \text{lambda}, \text{gradient}, \text{hessian}] =$ $\text{fmincon}(\dots)$	provides hessian value of f at the solution x

Limitations of fmincon :

1) It performs where the objective and constraint functions both are continuous and their first derivative is continuous.
2) When the problem is unsolvable, fmincon attempts to minimize the greatest constraint value.
3) It is not permitted when the upper and lower bounds are equal.

6.1.2 Comparing RMSE Values Across Models and Dates Using fmincon

Moneyness 0.9 to 1.1

After filtering the whole dataset, we obtain 7,704 entries for in-sample data and 7,558 entries for out-sample data within the moneyness range of 0.9 to 1.1. The following two tables provide insights into RMSE comparisons across various dates, computed by comparing Call Option Prices from our models (BS, VG, PBS, PVG sigma) and original market prices.

Table 6.1.1: Comparing RMSE values (In-Sample Data)

DATE	BS	VG	PBS	PVG SIGMA	NO. OF DATA
7-JAN	7.929665	4.785753	2.515547	2.448136	171
14-JAN	14.89131	5.357397	1.885703	1.87397	120
21-JAN	9.219772	9.21978	2.337538	2.306856	131
28-JAN	9.182866	3.158324	1.215337	1.20028	138
4-FEB	4.79375	2.145217	0.648331	0.64328	79
11-FEB	7.230624	7.230624	1.575918	1.535365	106
18-FEB	7.568333	2.033125	1.846168	1.834191	70
25-FEB	6.333845	2.135345	0.697327	0.696461	87
4-MAR	7.8079	2.559488	0.985444	0.983434	122
11-MAR	7.590019	2.509755	0.929807	0.927276	124
18-MAR	6.515619	6.515621	0.682802	0.679653	111
25-MAR	5.592051	3.359315	0.735778	0.733464	160
1-APR	8.194727	8.194729	0.654783	0.652945	134
8-APR	7.524948	7.524948	1.216137	1.144732	146
15-APR	9.89785	2.311349	0.593271	0.586806	145
22-APR	8.725876	8.725883	0.9951	0.992086	154
29-APR	4.862088	2.181061	0.780254	0.772902	165
6-MAY	6.07009	6.070092	0.982272	0.974275	144
13-MAY	7.071921	7.071482	0.768663	0.766982	126
20-MAY	4.625417	0.743255	0.61503	0.602786	86
27-MAY	5.132909	1.831737	0.752502	0.750435	98
3-JUN	7.121171	7.121178	0.640075	0.628988	181
10-JUN	5.101808	1.570981	0.992107	0.973907	158
17-JUN	5.48705	2.129262	0.619787	0.617132	145
24-JUN	5.200806	2.05173	0.777223	0.776371	160
1-JUL	5.153719	2.883474	1.043954	0.885183	124
8-JUL	6.086355	3.308328	1.229862	1.177807	118
15-JUL	6.248047	2.660828	1.33915	1.173964	105
23-JUL	5.16232	3.79726	1.27845	1.004268	96
29-JUL	5.729249	3.441215	1.246632	1.004863	120
5-AUG	4.868896	0.588273	0.82508	0.732341	115
12-AUG	4.575885	4.575888	0.930443	0.849071	120
19-AUG	6.139224	1.652194	0.717394	0.715695	149
26-AUG	5.194445	1.173193	0.971726	0.96152	140
2-SEP	7.701727	1.991546	0.92718	0.921287	174
9-SEP	5.481047	1.045519	1.326512	1.287979	155

16-SEP	1.404357	1.198256	0.757327	0.757132	200
23-SEP	4.971215	4.971215	1.29886	1.298555	216
30-SEP	5.744261	1.661181	0.802253	0.80014	200
7-OCT	5.663513	1.136046	1.048742	0.995561	206
14-OCT	4.441718	1.061486	0.874723	0.872904	184
21-OCT	4.419134	0.810595	0.83822	0.832635	197
28-OCT	4.646915	2.373897	1.097165	1.096926	200
4-NOV	6.440706	3.307254	2.373736	2.344956	191
11-NOV	5.314042	1.103658	1.407925	1.374151	186
18-NOV	3.132876	1.586359	1.136703	1.130151	181
2-DEC	5.221763	0.957261	1.214505	1.214302	170
9-DEC	2.798941	1.172846	2.202925	2.202886	240
16-DEC	2.252879	1.123525	1.670725	1.660863	205
23-DEC	3.75673	1.483549	2.590797	2.552288	229
30-DEC	2.196058	1.388399	2.12405	2.109557	222

Table 6.1.2 Comparing RMSE values (Out Sample Data)

DATE	BS	VG	PBS	PVG SIGMA	NO. OF DATA
8-JAN	10.452654	6.132996	4.925453	4.570485	130
15-JAN	15.057249	10.458811	13.859402	13.929786	120
22-JAN	9.219772	9.505613	2.340741	2.372752	124
29-JAN	8.176397	7.164626	8.358119	8.242952	117
5-FEB	5.825057	3.092138	3.131242	2.871655	93
12-FEB	5.944175	5.944168	2.617211	2.855185	65
19-FEB	8.185388	4.793882	4.828908	4.648216	79
26-FEB	8.004381	5.780546	6.10493	6.037337	111
5-MAR	13.448182	11.111376	14.16549	13.728631	126
12-MAR	10.17124	7.633459	5.846439	5.747004	110
19-MAR	12.03007	12.030259	14.213221	17.127393	145
26-MAR	4.599477	3.171361	1.858953	1.906348	158
2-APR	9.203664	9.203842	1.602437	1.581433	137
9-APR	8.827791	8.827722	1.901122	1.896911	133
16-APR	12.164353	10.523717	15.48517	15.538806	150
23-APR	9.142665	9.142646	1.291118	1.294389	157
30-APR	5.305617	2.671032	1.935724	1.851018	158
7-MAY	7.590549	7.590549	2.098116	2.032435	118
14-MAY	5.723105	5.722632	5.686569	5.551389	145
21-MAY	7.268674	9.769876	10.741546	10.410258	128
28-MAY	5.871895	3.213526	3.571545	3.500006	119
4-JUN	5.950547	5.950553	1.130267	1.195059	175
11-JUN	4.916367	2.244817	2.571463	2.456201	145

18-JUN	4.938253	3.165657	3.372702	3.417394	130
25-JUN	6.896295	10.345484	9.113312	9.081479	175
2-JUL	5.153719	3.171373	2.75362	2.522167	113
9-JUL	6.288132	3.995187	1.938953	1.946814	85
16-JUL	5.377444	4.660538	5.196981	5.488237	88
24-JUL	4.211614	3.04719	2.014902	2.223599	91
30-JUL	4.610602	2.84623	1.67876	1.947069	110
6-AUG	4.882413	1.205676	1.842631	1.583714	117
13-AUG	6.84619	6.846174	7.599142	7.61676	152
20-AUG	4.785188	3.132839	3.838204	3.845421	131
27-AUG	5.167202	1.177575	1.044386	1.083693	136
3-SEP	5.878816	2.616458	2.504922	2.517947	162
10-SEP	6.63646	1.67365	2.903792	2.584188	154
17-SEP	2.799378	2.559204	2.134547	2.12598	204
24-SEP	6.992624	6.992077	8.464388	8.459698	203
1-OCT	13.234318	12.584826	16.991628	16.744586	204
8-OCT	12.04001	12.054425	15.1765	14.801981	178
15-OCT	7.06691	5.487905	6.271403	6.242745	159
22-OCT	9.857896	10.019512	12.065528	11.981861	202
29-OCT	4.393106	5.099441	5.928011	5.917831	196
5-NOV	8.465765	6.832926	8.468478	8.405653	191
12-NOV	5.790684	1.870612	3.26601	3.059908	191
19-NOV	5.089314	5.030511	5.368839	5.421356	200
3-DEC	4.753742	1.341988	2.089622	2.091244	162
10-DEC	2.885963	1.32553	2.011184	2.011161	231
17-DEC	5.26234	4.565484	4.193294	4.257392	211
24-DEC	5.488911	5.847378	6.06528	5.610359	225
31-DEC	7.466038	8.102038	8.17563	7.931438	214

For In-sample data maximum RMSE= 14.891307 (found in BS on 14 Jan) and minimum RMSE= 0.586806 (found in PVG sigma on 15 April).
For the Out-Sample data, RMSE (max) = 17.127393 (found in PVG sigma on 4 Jun) and RMSE (min) = 1.044386 (found in PBS on 29 Oct).

Utilizing the data presented in the tables above, we created the following graphs to enhance our understanding of how RMSE values are compared in different models.

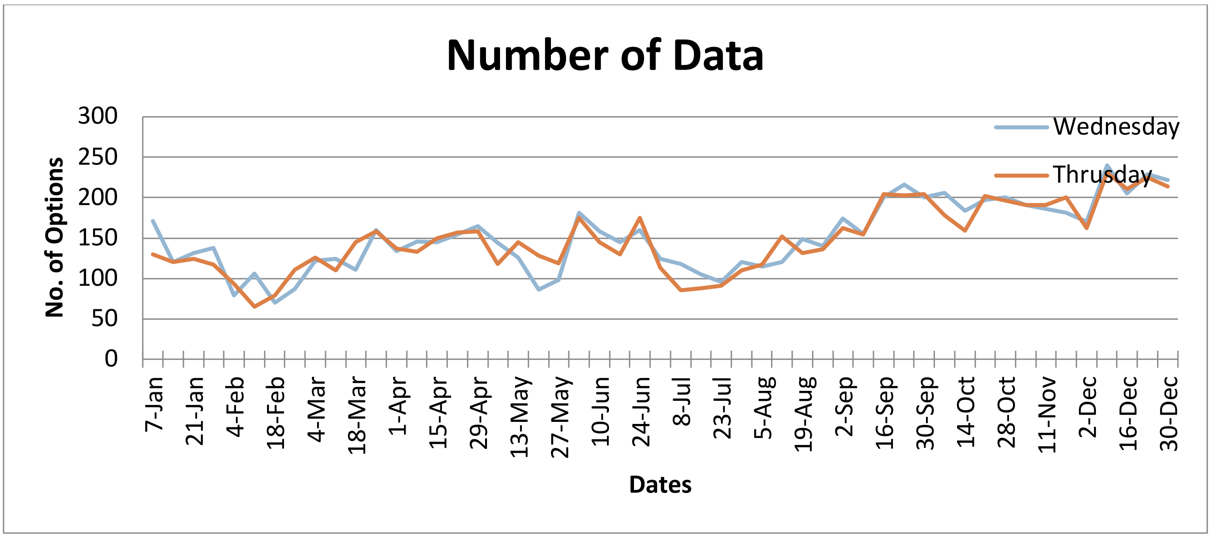

Figure 6.1.1: <u>No. of Data used for Calibration in the Moneyness range .9 to 1.1</u>

Figure 6.1.1 illustrates the count of options post-dataset filtration across various dates. Dates are displayed on the x-axis, while the number of options utilized in calibration is depicted on the y-axis.

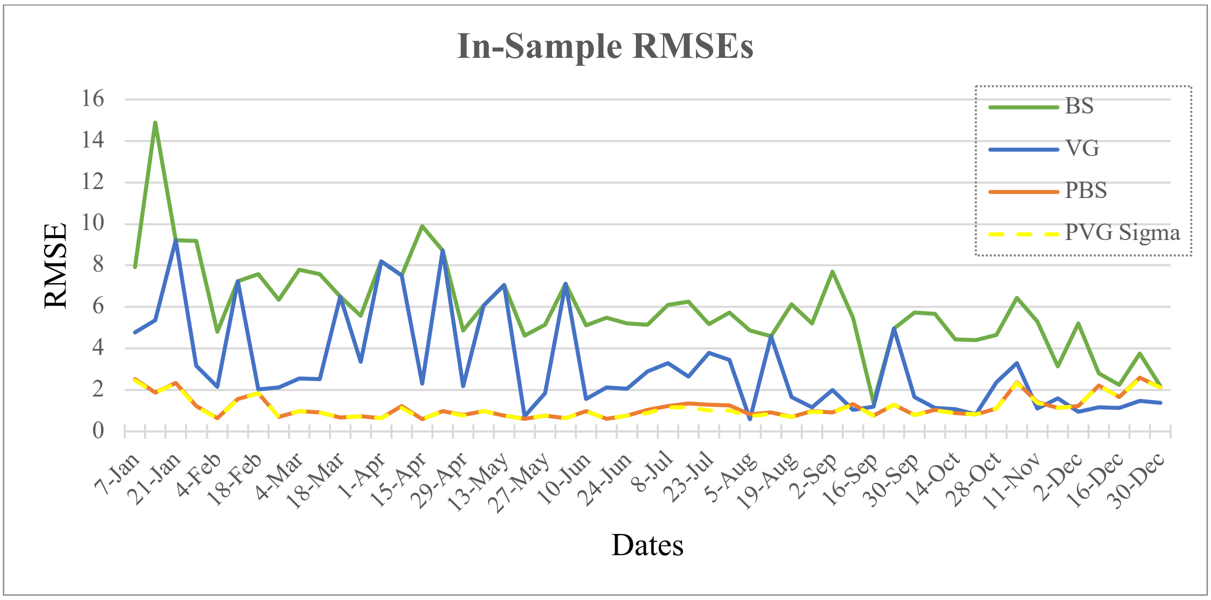

Figure 6.1.2: <u>RMSE Values Across Different Dates for In-Sample Data Using Discussed Models</u>

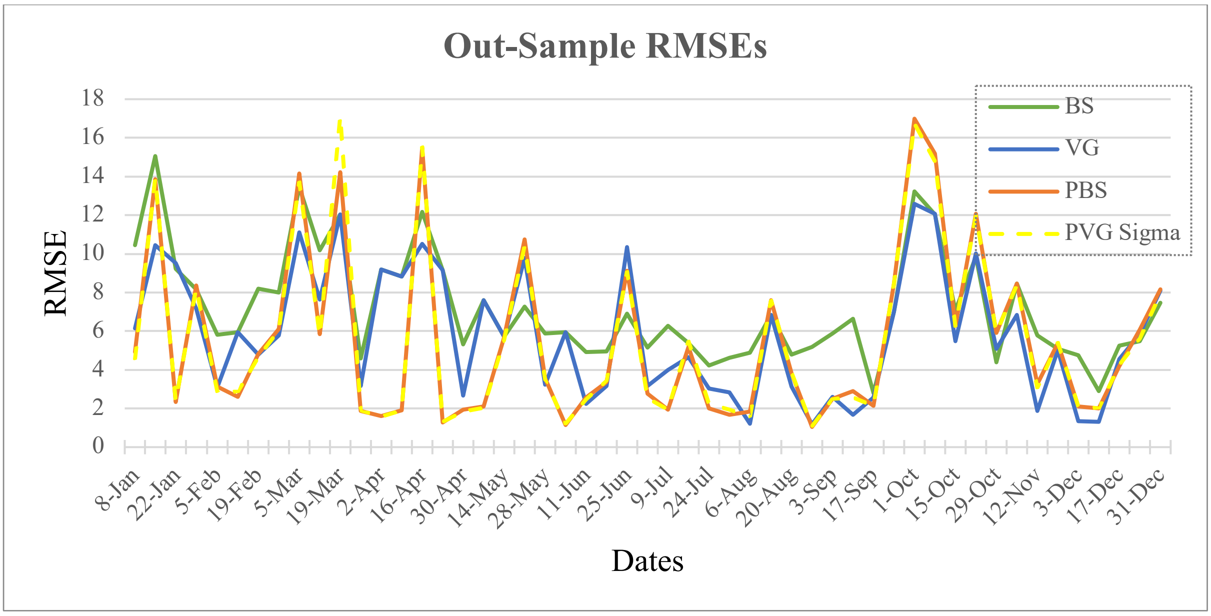

Figure 6.1.3: RMSE Values Across Different Dates for Out-Sample Data Using Discussed Models

In Figures 6.1.2 and 6.1.3, the RMSE values are displayed across different dates, with dates represented on the x-axis and RMSE values on the y-axis. Different colored lines correspond to different models.

From the graphical representations, it is evident that PVG and PBS exhibited nearly identical performance in both in-sample and out-sample data. Conversely, the other two models, BS and VG, demonstrated weaker performance in the in-sample dataset but exhibited improvement in the out-sample data, with VG showing particularly noteworthy improvement.

Overall Performance:

Table 6.1.3: Model Comparison: RMSE Statistics (In Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	5.9689889	5.592051	0.00%	0.00%	19.61%	**80.39%**
VG	3.15667992	2.181061	17.65%	0.00%	**68.63%**	19.61%
PBS	1.17094006	0.992107	0.00%	**82.35%**	17.65%	0.00%
PVG	1.13842545	0.974275	**82.35%**	17.65%	0.00%	0.00%

In the In-Sample data, PVG has the lowest mean and median RMSE, indicating the best predictive accuracy. It achieves the 1st best result 82.35% of the time, demonstrating consistent performance. PBS has the second-lowest mean and median RMSE in In-Sample data. It achieves the 2nd best result 82.35% of the time. VG performs well in In-Sample data with a relatively low mean and median RMSE. It achieves the 3rd best result 68.63% of the time, demonstrating consistency in performance.
BS has the highest mean and median RMSE in In-Sample data, suggesting lower predictive accuracy.

Table 6.1.4: Model Comparison: RMSE Statistics (Out Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	7.18310973	6.288132	15.69%	13.73%	11.76%	58.82%
VG	5.86819676	5.722632	41.18%	13.73%	37.25%	9.80%
PBS	5.54387912	3.838204	21.57%	27.45%	29.41%	21.57%
PVG	5.53465341	3.845421	21.57%	43.14%	23.53%	9.80%

In the Out-Sample data, PVG exhibits the lowest mean RMSE, while PBS boasts the lowest median RMSE, indicating that different models excel in different aspects. Examining the percentage of times each model achieves the 1st, 2nd, 3rd, 4th best results, VG stands out by achieving the 1st best result 41.18% of the time and the 3rd best result 37.25% of the time, making it the top-performing model in the out-sample data. It is followed by PVG sigma, which achieves the 1st best result 21.57% of the time, the 2nd best result 43.14% of the time, and the 3rd best result 23.53% of the time. PBS also demonstrates a strong performance with 1st best results occurring 21.57% of the time, 2nd best results 27.45% of the time, and 3rd best results 29.41% of the time. In contrast, BS consistently ranks 4th, with the highest RMSE values in 58.82% of the cases.

Overall, PVG emerges as the top-performing model, consistently delivering the lowest RMSE values in both In-Sample and Out-Sample datasets. VG and PBS also exhibit competitive performance, while BS consistently lags behind with the highest RMSE values.

Moneyness .85 to 1.15:

After filtering the whole dataset, we obtained 10,455 entries for in-sample data and 10,543 entries for out-sample data within the moneyness range of 0.85 to 1.15. The following tables, akin to Table 6.1.1 and 6.1.2, offer a comprehensive overview of RMSE comparisons across different dates.

Table 6.1.5: Comparing RMSE values (In Sample Data)

DATE	BS	VG	PBS	PVG SIGMA	NO. OF DATA
7-JAN	7.727031	5.882251	3.207348	3.182252	247
14-JAN	13.882387	6.481968	2.22386	2.187569	178
21-JAN	8.957758	6.189885	2.774667	2.714299	193
28-JAN	8.353852	3.861716	1.386714	1.348055	204
4-FEB	6.118501	2.71524	0.867336	0.774609	118
11-FEB	7.230624	7.230624	1.575918	1.535365	148
18-FEB	8.372774	8.372774	1.708168	1.691633	100
25-FEB	7.005387	2.401325	0.773312	0.768606	121
4-MAR	8.078014	7.761572	1.196398	1.192621	165
11-MAR	6.79066	2.812868	1.293803	1.211422	164
18-MAR	6.592709	2.98909	1.044698	1.033452	149
25-MAR	5.42708	4.070682	0.710669	0.710622	231

1-APR	8.557171	3.141251	0.770351	0.761799	176
8-APR	7.193523	7.19353	1.319276	1.27048	202
15-APR	8.933117	3.050798	0.837434	0.82699	222
22-APR	7.983533	7.983533	1.071495	1.064803	219
29-APR	4.442449	2.727912	0.737572	0.737143	243
6-MAY	5.948083	1.726114	1.249852	1.182203	199
13-MAY	7.068732	2.759153	0.88624	0.879064	175
20-MAY	4.751831	1.059195	0.849329	0.831016	137
27-MAY	5.833711	2.211408	1.112606	1.103816	157
3-JUN	6.817494	6.817501	1.090427	0.985855	242
10-JUN	5.104293	1.868379	2.207747	1.876223	218
17-JUN	5.692548	2.404262	1.119288	1.097206	194
24-JUN	6.476234	3.4797	2.31929	2.30096	210
1-JUL	5.062436	3.172689	1.404773	1.174449	173
8-JUL	5.707733	3.542826	2.436782	2.221948	168
15-JUL	6.507399	6.507412	1.888811	1.705487	144
22-JUL	4.628579	3.603553	1.234038	0.982438	139
29-JUL	5.341968	3.473727	1.229261	1.010043	169
5-AUG	4.803594	0.900164	1.156029	1.1326	162
12-AUG	4.82447	0.989319	2.485792	2.429147	176
19-AUG	6.309595	1.839227	0.93705	0.912023	203
26-AUG	6.204883	1.442873	1.288035	1.167157	190
2-SEP	7.808332	2.209855	1.248322	1.123653	232
9-SEP	6.548982	1.241257	2.332619	2.285918	208
16-SEP	1.463244	1.321143	1.075529	1.065656	264
23-SEP	4.564328	4.564328	1.786973	1.766976	293
30-SEP	5.200678	1.612931	1.363729	1.345897	265
7-OCT	5.239561	1.249598	1.576336	1.467107	252
14-OCT	4.03149	0.980656	0.957818	0.935335	231
21-OCT	4.040558	0.898995	1.159643	1.098295	242
28-OCT	4.203036	2.604343	1.389662	1.369325	260
4-NOV	5.803745	3.487013	2.876878	2.836681	251
11-NOV	4.909472	1.3131	3.038929	2.925882	235
18-NOV	2.837819	1.674677	1.191408	1.183851	232
2-DEC	5.125657	1.10836	2.006126	1.680003	235
9-DEC	2.85454	1.32464	3.5803	3.507543	305
16-DEC	2.262624	1.21524	2.318079	2.28955	264
23-DEC	3.766373	1.362384	2.688737	2.668293	286
30-DEC	2.262624	1.21524	2.318079	2.28955	264

Table 6.1.6: Comparing RMSE values (Out Sample Data)

DATE	BS	VG	PBS	PVG SIGMA	NO. OF DATA
8-JAN	9.958726	7.114621	5.465235	5.773246	191
15-JAN	14.775827	11.309126	13.340557	13.483315	181
22-JAN	9.34219	6.653229	2.80424	2.800572	193
29-JAN	7.691434	6.908658	7.879764	7.680675	164
5-FEB	7.065063	3.967687	3.273581	3.287875	135
12-FEB	5.944175	5.944168	2.617211	2.855185	93
19-FEB	8.784505	8.784478	4.70569	4.506039	109
26-FEB	7.528908	5.980135	5.873336	5.802282	154
5-MAR	12.825319	12.818106	13.585243	13.271449	188
12-MAR	9.957339	7.78682	9.051514	8.983729	165
19-MAR	11.361954	14.159877	13.832732	13.38102	212
26-MAR	4.809792	4.345986	1.932131	1.938487	240
2-APR	9.056465	4.139863	1.451228	1.396513	190
9-APR	8.660674	8.660665	2.532369	2.251132	178
16-APR	11.741913	10.47534	14.247069	14.314771	213
23-APR	8.430708	8.430659	1.410715	1.360084	234
30-APR	4.772101	2.964919	1.782664	1.762716	237
7-MAY	7.304307	2.714903	2.164873	1.945447	168
14-MAY	6.875471	4.778623	5.991937	5.844801	210
21-MAY	7.493375	9.459156	10.099756	9.77075	183
28-MAY	5.51558	3.932826	3.891963	3.765115	172
4-JUN	6.44632	6.446298	1.18534	1.395395	239
11-JUN	5.411344	2.692181	4.307295	4.005036	196
18-JUN	5.788893	3.250866	3.014865	3.176879	182
25-JUN	6.671226	10.217174	10.009698	9.382002	246
2-JUL	5.222057	3.172689	2.41594	1.89743	279
9-JUL	6.320312	4.462108	3.001432	3.100717	124
16-JUL	5.316498	5.316468	4.880629	5.291639	132
23-JUL	4.300068	3.600663	1.884379	2.062598	134
30-JUL	5.424233	4.077041	2.473602	2.538547	165
6-AUG	4.872094	1.377556	1.904925	1.773915	161
13-AUG	7.149003	8.329177	10.619851	10.566772	197
20-AUG	6.222746	3.291154	3.374669	3.533147	188
27-AUG	6.220491	1.359047	1.176063	1.244126	184
3-SEP	7.012779	2.843168	2.090386	2.525246	229
10-SEP	6.105562	1.749383	3.62094	3.629335	198
17-SEP	2.692709	2.56282	2.380233	2.304336	285
24-SEP	6.745308	6.744486	8.851878	8.528472	288
1-OCT	12.031696	12.441904	15.659877	15.33557	313
8-OCT	10.945912	11.588556	13.872408	13.268913	256

15-OCT	6.649968	5.146874	5.527443	5.713474	207
22-OCT	9.318649	9.432984	11.448396	11.115857	254
29-OCT	4.293903	4.914507	5.815603	5.671972	257
5-NOV	7.883603	6.517039	7.741409	8.124909	250
12-NOV	6.400583	3.045481	4.95981	4.5152	203
19-NOV	4.850405	4.877093	4.973622	4.849488	252
3-DEC	5.726026	1.721043	2.385338	2.151878	224
10-DEC	3.099513	1.511309	3.468234	3.3369	239
17-DEC	5.036251	4.321876	4.219892	3.950833	272
24-DEC	5.05574	5.38513	5.372852	5.65327	291
31-DEC	9.604829	8.253255	8.43574	8.922378	288

For In-Sample data maximum RMSE = 13.882387 (found in BS on 14 Jan) and minimum RMSE = 0.710622 (found in PVG on 25 March).
For the Out-Sample data, maximum RMSE= 15.659877 (found in PBS on 1 Oct) and minimum RMSE = 1.176063 (found in PBS on 27 August).

Using the data sourced from table 6.1.5, 6.1.6, we have generated graphs to explore the comparative analysis of RMSE values, this time focusing moneyness .85 to 1.15.

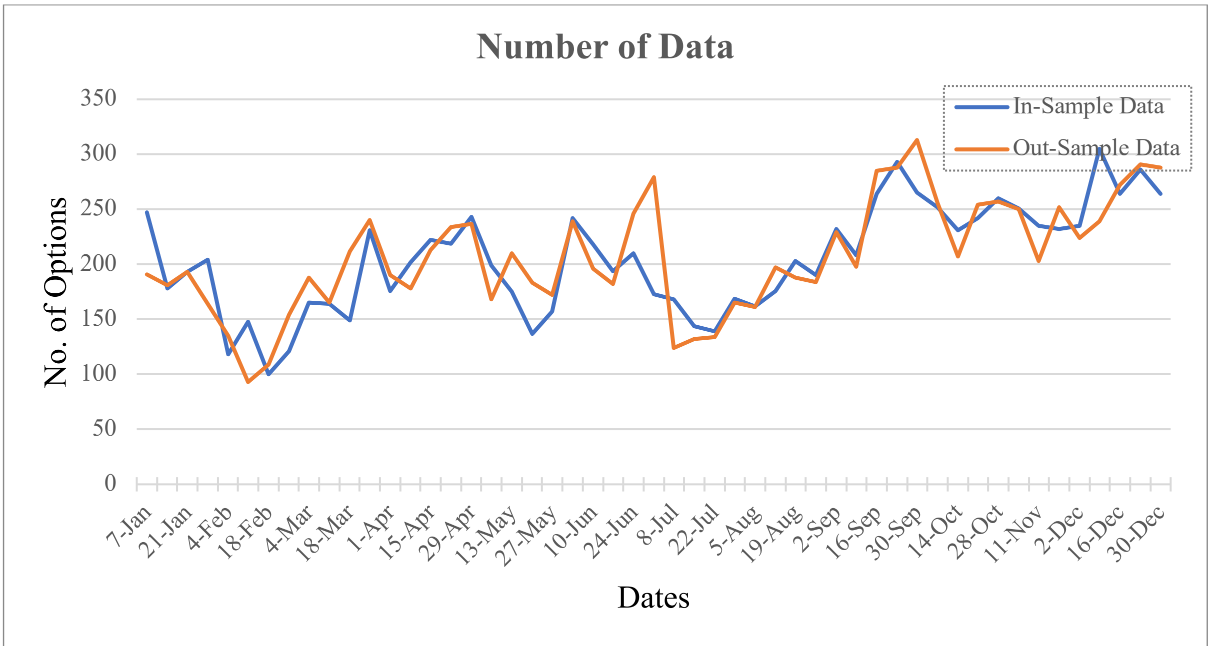

Figure 6.1.4: No. of Data used for Calibration the Moneyness range .85 to 1.15

Figure 6.1.4 visually represents the count of options following the dataset filtration process across a range of dates. The x-axis denotes the dates, while the y-axis displays the number of options used in the calibration.

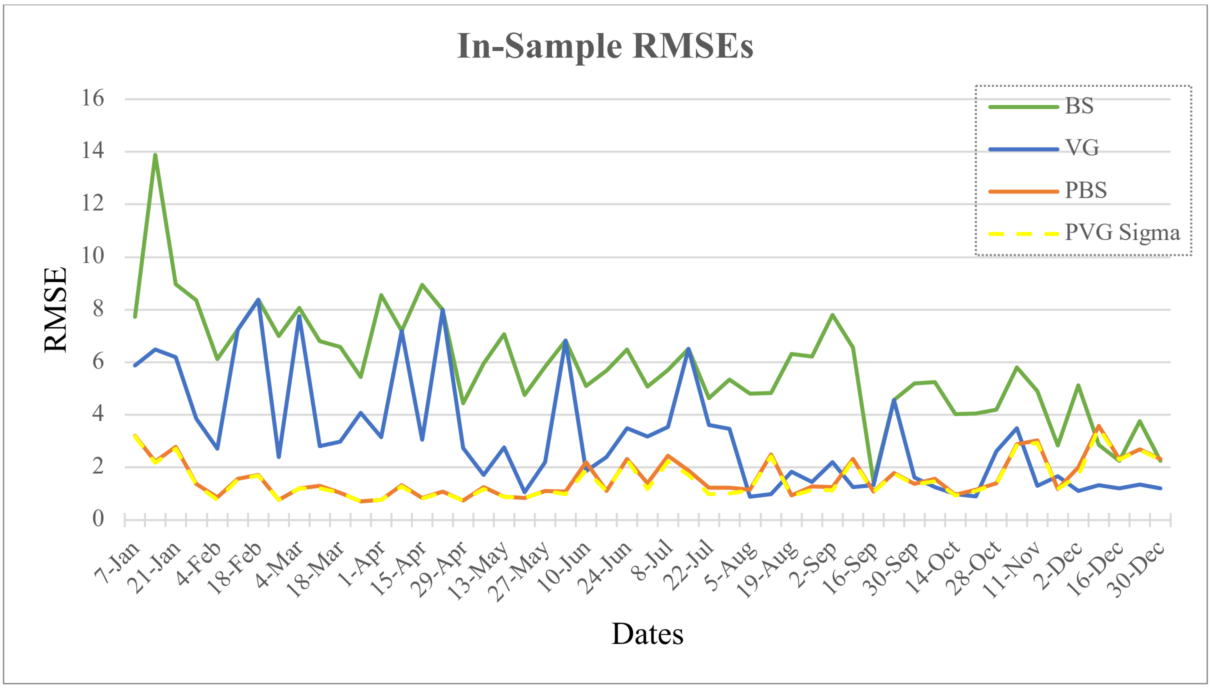

Figure 6.1.5: RMSE Values Across Different Dates for In-Sample Data Using Discussed Models

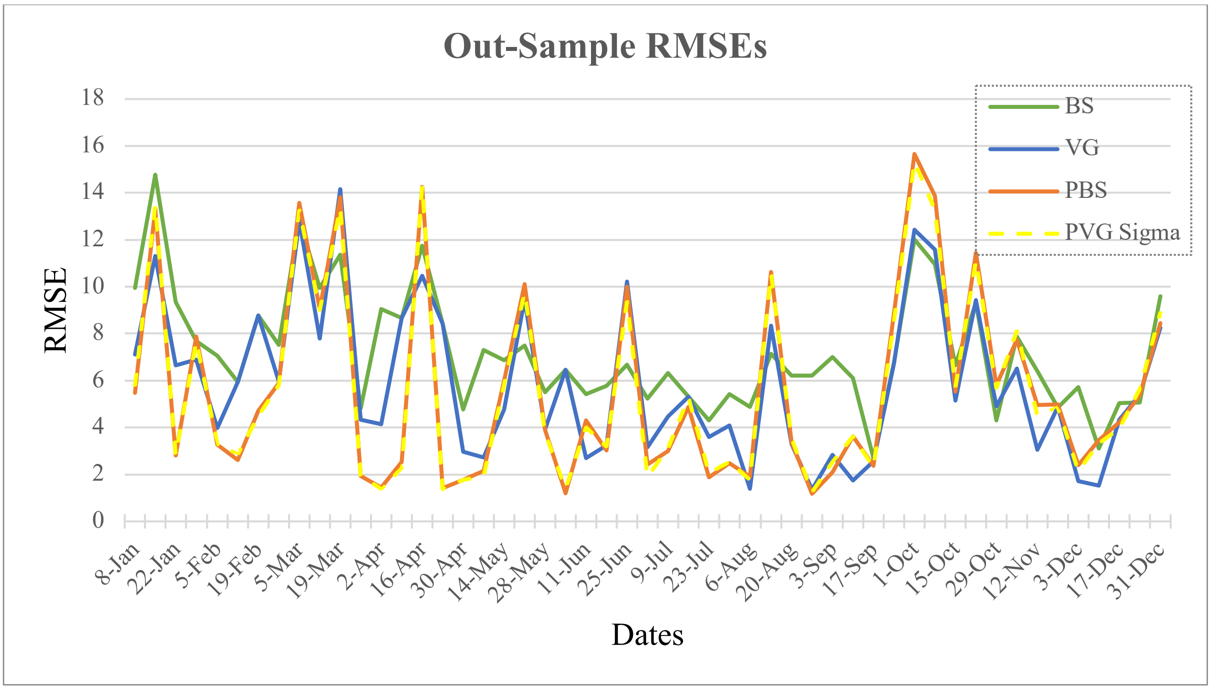

Figure 6.1.6: RMSE Values Across Different Dates for Out-Sample Data Using Discussed Models

In Figures 6.1.5 and 6.1.6, the RMSE values are displayed across different dates, with dates represented on the x-axis and RMSE values on the y-axis. Different colored lines correspond to different models.

Based on the graphical depictions, it appears that the performance trends are akin to the RMSEs observed within the moneyness range of 0.9 to 1.1.

Overall Performance:

Table 6.1.7: Model Comparison: RMSE Statistics (In Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	5.91476894	5.803745	0.00%	3.92%	13.73%	**82.35%**
VG	3.17741727	2.71524	21.57%	0.00%	**64.71%**	13.73%
PBS	1.59418698	1.293803	0.00%	**78.43%**	17.65%	3.92%
PVG	1.52633078	1.192621	**78.43%**	17.65%	3.92%	0.00%

Table 6.1.8: Model Comparison: RMSE Statistics (Out Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	7.229697	6.671226	17.65%	9.80%	3.92%	**68.63%**
VG	5.92116029	5.146874	**31.37%**	11.76%	**52.94%**	3.92%
PBS	5.66679524	4.307295	25.49%	35.29%	17.65%	21.57%
PVG	5.60277327	4.005036	25.49%	**43.14%**	25.49%	5.88%

We rank the models based on the provided in-sample and out-sample tables, which include mean, median, and the percentage of times each model achieves the 1st, 2nd, 3rd, and 4th best results.

Despite some variations in the statistical metrics, the overall performance continues to follow a similar trend as observed in our previous results within the moneyness range of 0.9 to 1.1.

6.2 Model Evaluation Using ‘lsqnonlin’

6.2.1 Introduction of MATLAB function ‘lsqnonlin’

‘lsqnonlin’ is a MATLAB command that solves nonlinear least-squares (nonlinear data-fitting) problems. ‘lsqnonlin’ solves nonlinear least-squares curve fitting problems of the form -

$$\min_x \|f(x)\|_2^{\,2} = \min_x \left(f1(x)^2+f2(x)^2+\ldots+fn(x)^2\right)$$

subject to the constraints

$$f(x) = \begin{cases} c(x) \le 0, c_{eq}(x) = 0 \\ Ax \le b, \quad A_{eq}x = c_{eq} \\ l_b \le x \le u_b \end{cases}$$

Here x, l_b, and u_b can be vectors or matrices.

lsqnonlin do not take the objective function as the scalar value $\|f(x)\|_2{}^2$ (the sum of squares). lsqnonlin requires the objective function to be the *vector*-valued function such as,

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ \vdots \\ \vdots \\ f_n(x) \end{pmatrix}$$

Syntax:

Equation Distribution	Description
Input Based Arguments	
1)$x = \mathrm{lsqnonlin}(f, x_0)$	minimizing the sum of the square of the function fun which gives vector values but not sum of the squares of the values.
2)$x = \mathrm{lsqnonlin}(f, x_0, l_b, u_b)$	solving the problem when the function expressed on a range.
3)$x = \mathrm{lsqnonlin}(f, x_0, l_b, u_b, \mathrm{operation})$	minimizing the optimization problem.
4)$x = \mathrm{lsqnonlin}(\mathrm{problem})$	finds minimum for a problem
Output Based Arguments	
5)$[x, \mathrm{resnorm}] = \mathrm{lsqnonlin}(\dots)$	gives the residual x's squared two norm value.
6)$[x, \mathrm{resnorm}, \mathrm{residual}] = \mathrm{lsqnonlin}(\dots)$	provides the residual function value at the solution x.
7)$[x, \mathrm{resnorm}, \mathrm{residual}, \mathrm{exitflag}, \mathrm{output}] = \mathrm{lsqnonlin}(\dots)$	giving a structure output of the optimization.
8)$[x, \mathrm{resnorm}, \mathrm{residual}, \mathrm{exitflag}, \mathrm{output}, \mathrm{lambda}] = \mathrm{lsqnonlin}(\dots)$	provides a formation $lambda$ which contains Lagrange multiplier at the solution x
9) $[x, \mathrm{resnorm}, \mathrm{residual}, \mathrm{exitflag}, \mathrm{output}, \mathrm{lambda}, \mathrm{jacobian}] = \mathrm{lsqnonlin}(\dots)$	returns a $jacobian$ value of the function at the soltion x.

6.2.2 Comparing RMSE Values Across Models and Dates Using lsqnonlin

<u>Moneyness .9 to 1.1</u>

After filtering the entire dataset, we've extracted 7,704 entries for in-sample data and 7,558 entries for out-sample data, all falling within the moneyness range of 0.9 to 1.1. The following two tables, akin to Section 6.1.2, offer a detailed analysis of RMSE comparisons across various dates. These comparisons are derived by evaluating Call Option Prices using our models (BS, VG, PBS, PVG sigma) and comparing them with the original market prices.

Table 6.2.1: Comparing RMSE values (In Sample Data)

DATE	*BS*	*VG*	*PBS*	*PVG SIGMA*	*NO. OF DATA*
7-JAN	7.929665	4.792114	2.515547	2.457295	171
14-JAN	14.891307	5.35742	1.885703	1.873457	120
21-JAN	9.219772	4.962475	2.337538	2.331646	131
28-JAN	9.182866	3.158219	1.215337	1.200869	138
4-FEB	4.79375	1.986189	0.648331	0.648532	79
11-FEB	6.150769	2.21068	0.599929	0.600219	153
18-FEB	7.568333	1.944409	1.846168	1.84614	70
25-FEB	6.333845	1.955953	0.697327	0.697085	87
4-MAR	7.8079	2.561062	0.985444	0.98577	122
11-MAR	7.590019	2.516691	0.929807	0.928905	124
18-MAR	6.515619	2.418728	0.682802	0.682878	111
25-MAR	5.592051	3.356757	0.735778	0.733572	160
1-APR	8.194727	2.31299	0.654783	0.654781	134
8-APR	7.524948	1.708846	1.216137	1.214311	144
15-APR	9.89785	2.613149	0.593271	0.586767	145
22-APR	8.725876	1.696372	0.9951	0.994367	154
29-APR	4.862088	2.13974	0.780254	0.77277	165
6-MAY	6.07009	1.242903	0.982272	0.982128	144
13-MAY	7.071921	2.287429	0.768663	0.768942	126
20-MAY	4.625417	0.723007	0.61503	0.612494	86
27-MAY	5.132909	1.675386	0.752502	0.750316	98
3-JUN	7.121171	2.235698	0.640075	0.637766	181
10-JUN	5.101808	1.516163	0.992107	0.99059	158
17-JUN	5.48705	2.015501	0.619787	0.619241	145
24-JUN	5.200806	1.939848	0.777223	0.776314	160

1-JUL	4.420008	0.995831	0.773096	0.773444	122
8-JUL	4.972261	1.308125	1.371982	1.371495	116
15-JUL	5.416914	0.832606	0.676631	0.673076	103
23-JUL	4.824849	0.584678	0.609013	0.604382	94
29-JUL	5.070356	0.848931	0.655685	0.640827	118
5-AUG	4.868896	0.59723	0.82508	0.787448	115
12-AUG	4.575885	0.723675	0.930443	0.928555	120
19-AUG	6.139224	1.584246	0.717394	0.712955	149
26-AUG	5.194445	1.09413	0.971726	0.960234	140
2-SEP	7.701727	1.922872	0.92718	0.918444	174
9-SEP	5.481047	1.019213	1.326512	1.323314	155
16-SEP	1.404357	0.947392	0.757327	0.757127	200
23-SEP	4.971215	1.937535	1.29886	1.290277	216
30-SEP	5.744261	1.490348	0.802253	0.803055	200
7-OCT	5.663513	1.165643	1.048742	1.046004	206
14-OCT	4.441718	0.842253	0.874723	0.872021	184
21-OCT	4.419134	0.837216	0.83822	0.837598	197
28-OCT	4.646915	2.388711	1.097165	1.093573	200
4-NOV	6.440706	3.305812	2.373736	2.345205	191
11-NOV	5.314042	1.126585	1.407925	1.340198	186
18-NOV	3.132876	1.586327	1.136703	1.130926	181
2-DEC	5.221763	1.100449	1.214505	1.20948	170
9-DEC	2.798941	1.172846	2.202925	2.200653	240
16-DEC	2.252879	1.123525	1.670725	1.665207	205
23-DEC	3.75673	1.483549	2.590797	2.576261	229
30-DEC	2.196058	1.388399	2.12405	2.122722	222

Table 6.2.2: Comparing RMSE values (Out Sample Data)

DATE	*BS*	*VG*	*PBS*	*PVG SIGMA*	*NO. OF DATA*
8-JAN	10.452644	6.178207	4.925453	4.625647	130
15-JAN	15.057252	10.458572	13.859402	13.927467	120
22-JAN	9.50559	5.080301	2.340741	2.339302	124
29-JAN	8.176401	7.178403	8.358119	8.242561	117
5-FEB	5.825057	3.037798	3.131242	3.1282	93
12-FEB	5.448981	3.556203	3.322856	3.321952	119
19-FEB	8.185385	4.639582	4.828908	4.827184	79
26-FEB	8.004383	5.805093	6.10493	6.102452	111
5-MAR	13.448181	11.044611	14.16549	14.165158	126
12-MAR	10.171239	7.558635	5.846439	5.847033	110

19-MAR	12.03007	14.245181	14.213221	14.210782	145
26-MAR	4.599479	3.15314	1.858953	1.904717	158
2-APR	9.203663	3.409565	1.602437	1.602906	137
9-APR	8.827791	2.470454	1.901122	1.896911	133
16-APR	12.164356	10.385672	15.48517	15.538464	150
23-APR	9.142666	2.275668	1.291118	1.291112	157
30-APR	5.305616	2.627719	1.935724	1.855088	158
7-MAY	7.590549	2.309045	2.098116	2.097772	118
14-MAY	5.723105	4.215793	5.686569	5.678083	145
21-MAY	7.268689	9.789992	10.741546	10.690034	128
28-MAY	5.871882	3.215523	3.571545	3.545727	119
4-JUN	5.950547	1.94679	1.130267	1.118839	175
11-JUN	4.916366	2.121583	2.571463	2.559708	145
18-JUN	4.938253	3.15063	3.372702	3.373675	130
25-JUN	6.896295	10.635984	9.113312	9.081479	175
2-JUL	4.621455	1.784867	2.685535	2.682537	112
9-JUL	5.763122	1.848563	2.205299	2.206288	84
16-JUL	5.130108	4.496266	5.825856	5.811482	87
24-JUL	3.915241	0.687664	2.28353	2.291337	90
30-JUL	4.130046	1.228387	1.792107	1.803371	109
6-AUG	4.882416	1.201723	1.842631	1.839845	117
13-AUG	6.846197	8.10774	7.599142	7.543149	152
20-AUG	4.785186	3.101215	3.838204	3.795058	131
27-AUG	5.167202	1.121396	1.044386	1.012159	136
3-SEP	5.878816	2.551746	2.504922	2.433362	162
10-SEP	6.202025	1.578036	2.518864	2.501746	152
17-SEP	2.799378	2.329726	2.134547	2.140428	204
24-SEP	6.992625	6.596672	8.464388	8.478983	203
1-OCT	13.23432	12.600869	16.991628	16.977481	204
8-OCT	12.040007	12.051794	15.1765	15.161565	178
15-OCT	7.066914	5.396668	6.271403	6.263593	159
22-OCT	9.857891	10.030479	12.065528	12.059568	202
29-OCT	4.393113	5.10446	5.928011	5.923291	196
5-NOV	8.465751	6.807016	8.468478	8.406235	191
12-NOV	6.183238	2.683826	3.287542	3.285329	205
19-NOV	5.089315	5.023105	5.368839	5.387764	200
3-DEC	4.753742	1.334454	2.089622	2.096869	162
10-DEC	2.885964	1.32553	2.011184	2.010298	231
17-DEC	5.26234	4.565483	4.193294	4.217352	211
24-DEC	5.488897	5.847374	6.06528	5.946846	225
31-DEC	7.466035	8.102039	8.17563	8.168501	214

For In-Sample data maximum RMSE= 14.891307 (found in BS on 14 Jan) and minimum RMSE= 0.584678 (found in VG on 23 July).

For the Out-Sample data, maximum RMSE 16.991628 (found in PBS on 1 Oct) and minimum RMSE = 0.687664 (found in VG on 24 July).

Utilizing the data presented in the tables above, we created the following graphs to enhance our understanding of how RMSE values are compared in different models.

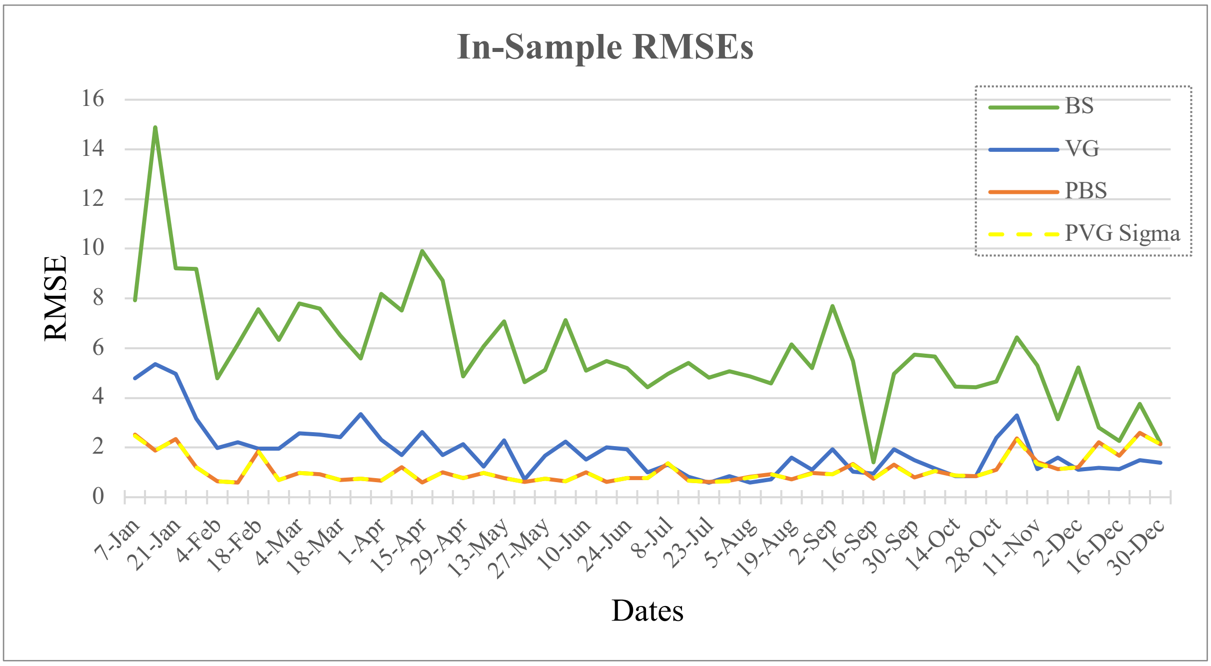

Figure 6.2.1: <u>RMSE Values Across Different Dates for In-Sample Data Using Discussed Models</u>

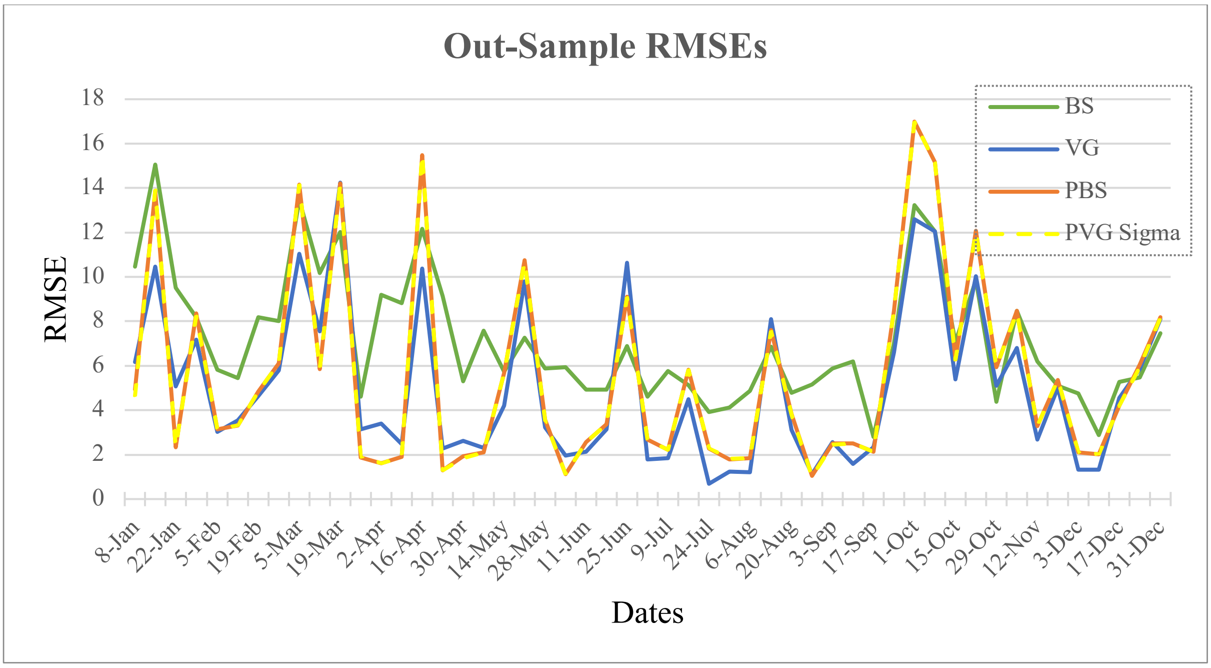

Figure 6.2.2: <u>RMSE Values Across Different Dates for Out-Sample Data Using Discussed Models</u>

In Figures 6.2.1 and 6.2.2, we present a visual representation of RMSE values plotted across different dates. The x-axis represents the dates, while the y-axis displays the RMSE values. Each model is depicted by a distinct colored line.
These graphs reveal that PVG and PBS performed remarkably similarly and good performance in both the in-sample and out-sample datasets, mirroring the outcomes achieved during calibrations using 'fmincon'. The other two models, BS and VG, initially demonstrated weaker performance in the in-sample dataset but displayed substantial improvements in the out-sample data. Notably, VG exhibited significant enhancements, paralleling our previous 'fmincon' calibrations.

Overall Performance:

Table 6.2.3: Model Comparison: RMSE Statistics (In Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	5.87575053	5.416914	0.00%	0.00%	0.00%	100.00%
VG	1.8575658	1.675386	23.53%	0.00%	74.51%	0.00%
PBS	1.11157476	0.929807	13.73%	60.78%	25.49%	0.00%
PVG	1.10454188	0.928555	60.78%	39.22%	0.00%	0.00%

The findings presented here align closely with our earlier results obtained using 'fmincon,' albeit with varying statistical measures. In the In-Sample data, PVG stands out with the lowest mean and median RMSE, signifying superior predictive accuracy. It secures the 1st best result 60.78% of the time and the 2nd best result 39.22% of the time, demonstrating consistent excellence. Following closely, PBS exhibits the second-lowest mean and median RMSE in the In-Sample data, achieving the 2nd best result 60.78% of the time. VG consistently attains the 3rd best result, doing so 74.51% of the time. Conversely, BS registers the highest mean and median RMSE values and consistently delivers the 4th best result in In-Sample data, indicating relatively poorer predictive accuracy.

Table 6.2.4: Model Comparison: RMSE Statistics (Out Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	7.13736831	6.183238	17.65%	13.73%	1.96%	66.67%
VG	5.05876945	4.215793	52.94%	11.76%	29.41%	5.88%
PBS	5.57429794	3.838204	9.80%	31.37%	37.25%	21.57%
PVG	5.55719	3.795058	19.61%	43.14%	31.37%	5.88%

In the Out-Sample data, VG exhibits the lowest mean RMSE, while PVG boasts the lowest median RMSE, indicating that different models excel in different aspects. Examining the percentage of times each model achieves the 1st, 2nd, 3rd, 4th best results, VG stands out by achieving the 1st best result 52.94% of the, making it the top-performing model in the out-sample data. It is followed by PVG sigma, which achieves the 1st best result 19.61% of the time, the 2nd best result 43.14% of the time, and the 3rd best result 31.37% of the time. PBS also demonstrates a strong performance with 2nd best results 31.37% of the time, and 3rd best results 37.25% of the time. In contrast, BS consistently ranks 4th, with the highest RMSE values in 66.67% of the cases.

Overall, PVG emerges as the top-performing model, consistently delivering the lowest RMSE values in both In-Sample and Out-Sample datasets. VG and PBS also exhibit competitive performance, while BS consistently lags behind with the highest RMSE values.

Moneyness .85 to 1.15

After filtering the whole dataset, we obtained 10,455 entries for in-sample data and 10,543 entries for out-sample data within the moneyness range of 0.85 to 1.15. The following tables, akin to Table 6.2.1 and 6.2.2, offer a comprehensive overview of RMSE comparisons across different dates.

Table 6.2.5: Comparing RMSE values (In Sample Data)

DATE	BS	VG	PBS	PVG SIGMA	NO. OF DATA
7-JAN	7.727031	5.673729	3.207348	3.194266	247
15-JAN	13.882387	6.457298	2.22386	2.187316	178
22-JAN	8.957758	6.113526	2.774667	2.749814	193
29-JAN	8.353852	3.788477	1.386714	1.347436	204
4-FEB	6.118501	2.475642	0.867336	0.867039	118
11-FEB	7.230624	2.545957	1.575918	1.574306	148
18-FEB	8.372774	2.269469	1.708168	1.707142	167
25-FEB	7.005387	2.221863	0.773312	0.773454	121
4-MAR	8.078014	3.162421	1.196398	1.196056	165
11-MAR	6.79066	2.709485	1.293803	1.293604	164
18-MAR	6.592709	2.777862	1.044698	1.045805	149
25-MAR	5.42708	4.030226	0.710669	0.709481	231
1-APR	8.557171	2.828621	0.770351	0.770311	176
8-APR	7.193523	2.005471	1.319276	1.318878	202
15-APR	8.933117	3.080375	0.837434	0.82674	222
22-APR	7.983533	1.938525	1.071495	1.071774	219
29-APR	4.442449	2.508185	0.737572	0.736997	243
6-MAY	5.948083	1.640922	1.249852	1.24988	199
13-MAY	7.068732	2.617633	0.88624	0.885891	175
20-MAY	4.751831	0.997426	0.849329	0.848742	137
27-MAY	5.833711	1.982645	1.112606	1.110436	157
3-JUN	6.817494	2.39141	1.090427	1.064015	242
10-JUN	5.104293	1.819357	2.207747	2.206765	218
17-JUN	5.692548	2.273171	1.119288	0.993568	194
24-JUN	6.476234	3.41701	2.31929	2.305193	210
1-JUL	4.571349	1.153571	1.552644	1.551337	170

8-JUL	4.899656	1.594143	2.724678	2.721875	165
15-JUL	5.996646	0.996021	0.833909	0.833812	141
23-JUL	4.380699	0.661609	0.937978	0.945241	136
29-JUL	4.809274	1.121495	0.830051	0.829108	166
5-AUG	4.803594	0.911105	1.156029	1.154005	162
12-AUG	4.82447	0.979921	2.485792	2.483747	176
19-AUG	6.309595	1.787872	0.93705	0.925382	203
26-AUG	6.204883	1.39498	1.288035	1.288537	190
2-SEP	7.808332	2.173634	1.248322	1.23011	232
9-SEP	6.548982	1.248856	2.332619	2.27976	208
16-SEP	1.463244	1.153994	1.075529	1.075964	264
23-SEP	4.564328	2.00278	1.786973	1.770068	293
30-SEP	5.200678	1.630679	1.363729	1.362885	265
7-OCT	5.239561	1.266335	1.576336	1.40046	252
14-OCT	4.034585	1.001313	0.97824	0.928092	233
21-OCT	4.117995	0.92201	1.212508	1.130695	246
28-OCT	4.242331	2.57569	1.128175	1.127338	279
4-NOV	5.766623	3.585108	3.562207	3.45146	292
11-NOV	4.909472	1.345634	3.038929	3.036516	235
18-NOV	3.340214	1.675147	1.216071	1.214567	252
2-DEC	5.125657	1.096661	2.006126	2.003206	235
9-DEC	2.871114	1.375615	3.450977	3.447518	339
16-DEC	2.253312	1.326029	1.86921	1.865832	303
23-DEC	3.691831	1.620401	2.764037	2.760409	334
30-DEC	2.108123	1.452732	2.580925	2.578672	309

Table 6.2.6: Comparing RMSE values (Out Sample Data)

DATE	**BS**	**VG**	**PBS**	**PVG SIGMA**	**NO. OF DATA**
8-JAN	9.958716	6.905261	5.465235	5.387332	191
16-JAN	14.775643	11.269904	13.340557	13.482456	181
23-JAN	9.342185	6.583909	2.80424	2.775654	193
30-JAN	7.691445	6.973977	7.879764	7.681366	164
5-FEB	7.065063	3.921116	3.273581	3.267479	135
12-FEB	5.944176	3.831126	2.617211	2.617209	93
19-FEB	8.784515	4.661787	4.70569	4.699344	161
26-FEB	7.528912	5.886038	5.873336	5.868211	154
5-MAR	12.825337	11.454713	13.585243	13.578999	188
12-MAR	9.957339	8.043135	9.051514	9.044731	165

19-MAR	11.361955	14.741959	13.832732	13.826932	212
26-MAR	4.809795	4.298628	1.932131	1.965415	240
2-APR	9.056468	3.876766	1.451228	1.452046	190
9-APR	8.660674	2.905787	2.532369	2.512071	178
16-APR	11.74195	10.32121	14.247069	14.314771	213
23-APR	8.430708	2.67136	1.410715	1.411534	234
30-APR	4.772101	2.79574	1.782664	1.765997	237
7-MAY	7.304307	2.620704	2.164873	2.165067	168
14-MAY	6.87547	4.737546	5.991937	5.977736	210
21-MAY	7.493388	9.497071	10.099756	10.061738	183
28-MAY	5.515593	3.893067	3.891963	3.876139	172
4-JUN	6.446314	2.373136	1.18534	1.164813	239
11-JUN	5.411346	2.638644	4.307295	4.304429	196
18-JUN	5.788854	3.226545	3.014865	2.977006	182
25-JUN	6.671257	10.308966	10.009698	9.983573	246
2-JUL	4.744076	1.929291	2.463128	2.460706	167
9-JUL	5.898573	2.069934	3.452595	3.452278	121
16-JUL	5.137368	4.800947	5.797402	5.802233	129
24-JUL	4.160902	0.817335	1.74864	1.756596	131
30-JUL	5.264303	1.579372	1.483818	1.49625	162
6-AUG	4.872134	1.371339	1.904925	1.89699	161
13-AUG	7.149109	8.232566	10.619851	10.610387	197
20-AUG	6.222688	3.290144	3.374669	3.385299	188
27-AUG	6.220481	1.324186	1.176063	1.181878	184
3-SEP	7.012733	2.803397	2.090386	2.079616	229
10-SEP	6.105565	1.720361	3.62094	3.615889	198
17-SEP	2.692709	2.320458	2.380233	2.378724	285
24-SEP	6.745309	6.761364	8.851878	8.998556	288
1-OCT	12.031695	12.423217	15.659877	15.647062	313
8-OCT	10.945917	11.595062	13.872408	17.654123	256
15-OCT	6.636136	5.145153	5.542703	5.670849	207
22-OCT	9.282477	9.766639	11.807885	11.79872	273
29-OCT	4.405849	5.136305	5.936407	5.934018	277
5-NOV	7.681643	6.795069	8.049019	8.101255	250
12-NOV	5.672065	2.760875	4.380971	4.377106	262
19-NOV	5.054651	4.916041	5.565978	5.561887	274
3-DEC	5.726025	1.758052	2.385338	2.388511	224
10-DEC	2.941345	1.471737	3.488109	3.482683	338
17-DEC	5.069652	4.360969	4.450416	4.448504	306
24-DEC	5.228603	5.353095	4.928535	4.914229	314
31-DEC	7.139703	7.764964	7.776281	7.775859	318

For in sample data RMSE (max) = 13.882387 (found in BS on 15 Jan) and RMSE (min) = 0.773312 (found in PBS on 25 Feb). For the out sample data, RMSE (max) = 17.654123 (found in PVG sigma on 8 Oct) and RMSE (min) = 0.817335 (found in VG on 24 July).

Using the data sourced from table 6.2.5, 6.2.6, we have generated graphs to explore the comparative analysis of RMSE values, this time focusing moneyness .85 to 1.15.

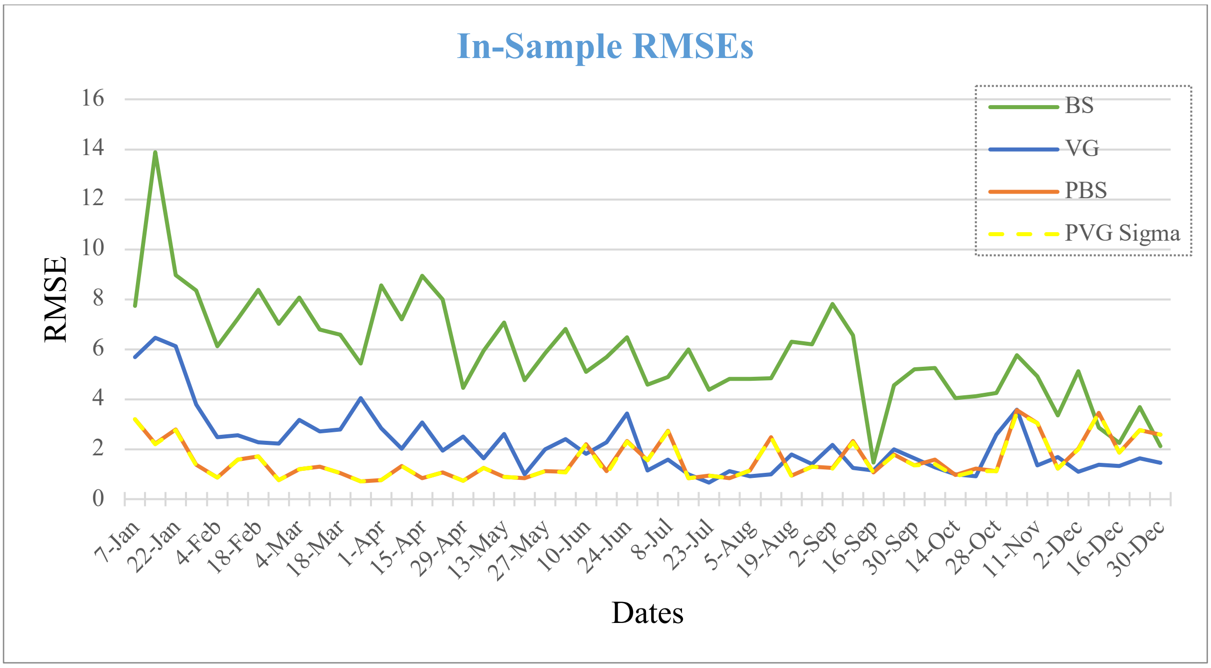

Figure 6.2.3: RMSE Values Across Different Dates for In-Sample Data Using Discussed Models

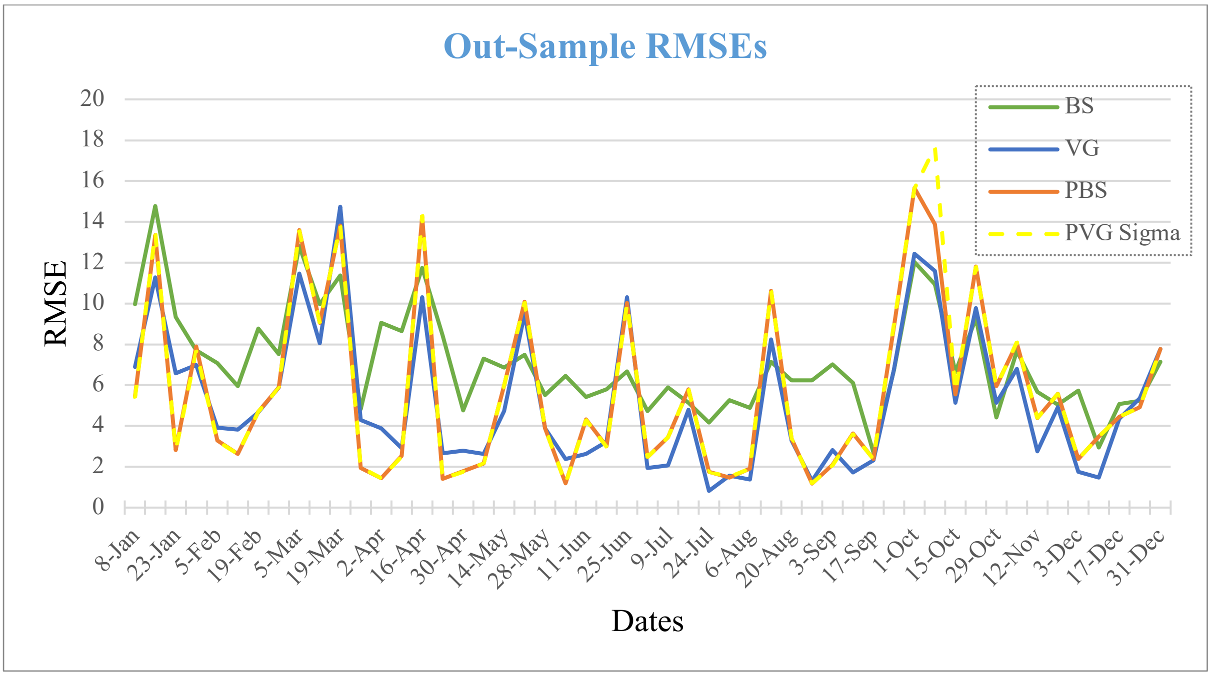

Figure 6.2.4: RMSE Values Across Different Dates for Out-Sample Data Using Discussed Models

In Figures 6.2.3 and 6.1.4, the RMSE values are displayed across different dates, with dates represented on the x-axis and RMSE values on the y-axis. Different colored lines correspond to different models.

Based on the graphical depictions, the performance trends appear to be similar to the RMSEs observed within the moneyness range of 0.9 to 1.1.

Overall Performance:

Table 6.2.7: Model Comparison: RMSE Statistics (In Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	5.8710989	5.766623	0.00%	3.92%	0.00%	96.08%
VG	2.19176551	1.938525	29.41%	0.00%	70.59%	0.00%
PBS	1.57393876	1.249852	11.76%	60.78%	23.53%	3.92%
PVG	1.55748049	1.24988	58.82%	35.29%	5.88%	0.00%

Table 6.2.8: Model Comparison: RMSE Statistics (Out Sample Data)

METHOD	MEAN	MEDIAN	1ST	2ND	3RD	4TH
BS	7.14225925	6.671257	19.61%	11.76%	3.92%	64.71%
VG	5.26874445	4.360969	45.10%	15.69%	33.33%	5.88%
PBS	5.67175414	4.380971	11.76%	33.33%	35.29%	19.61%
PVG	5.74573051	4.377106	23.53%	39.22%	27.45%	9.80%

Despite some variations in the statistical metrics, the overall performance continues to follow a similar trend as observed in our previous results within the moneyness range of 0.9 to 1.1.

6.3 Comprehensive Assessment of Model Performance

The consistent pattern across both optimization methods underscores the strength of the Practitioner Variance Gamma (PVG) model in predictive accuracy. PVG excelled by delivering the lowest RMSE values, making it the preferred choice for call option pricing within both of the specified moneyness ranges. PBS also demonstrated notable performance, particularly with the 'fmincon' method, ranking second in predictive accuracy. VG displayed competitive performance and showcased significant improvement in out-sample data. In contrast, the Black & Scholes (BS) model consistently lagged behind in predictive accuracy, consistently ranking last.

Despite some variations in statistical metrics between the two optimization methods, the overall trend in model performance remained consistent, offering valuable insights into the benefits of the PVG model in Option Pricing.

While the RMSE values for both in-sample and out-sample data were generally low, an interesting observation emerges when examining certain dates, particularly in the case of practitioner-based models (PBS and PVG). Given the inherent connection of practitioner models to implied volatility, we conducted a detailed analysis of implied volatility in both in-sample and out-sample datasets, focusing on instances with notable disparities in RMSE values.

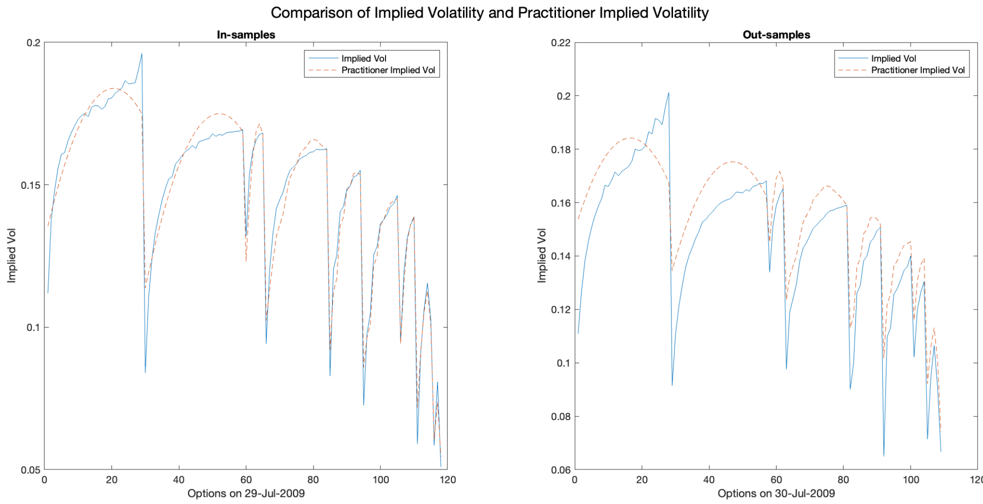

Figure 6.3.1: Comparison of Implied Volatility and Practitioner Implied Volatility on data from 29 and 30 July

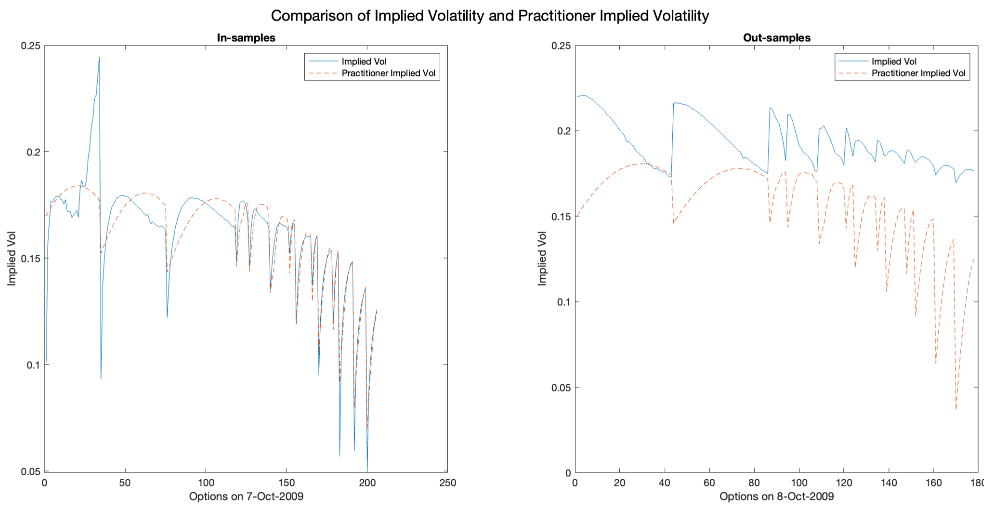

Figure 6.3.2: Comparison of Implied Volatility and Practitioner Implied Volatility on data from 7 and 8 October

Figure 6.3.1 illustrates the comparison between implied volatility and practitioner implied volatility on July 29th (in-sample) and July 30th (out-sample). Notably, the RMSE values for these dates exhibit minimal differences. Conversely, in Figure 6.3.2, which examines the implied volatility versus practitioner implied volatility on October 7th (in-sample) and October 8th (out-sample), a stark contrast emerges as the RMSE values display substantial disparities. These graphical representations shed light on a significant pattern: when the implied volatility of in-sample and out-sample data deviates, the practitioner's implied volatility in the out-sample data closely follows the in-sample trend. This phenomenon results in a noteworthy gap between actual and predicted implied volatility. Interestingly, this trend is less pronounced in cases where results are favorable, suggesting a potential connection between these discrepancies and unfavorable model outcomes.

Conclusion

This project marks a significant contribution to the field as it introduces the practitioner's version of the renowned Variance Gamma (VG) option pricing model into the literature for the first time. The application of the practitioner's approach to a technically complicated model like VG has yielded notably efficient results. It is essential to note that, when dealing with the VG model, the practitioner's approach surpasses the performance of the practitioner's approach applied to the typical Black-Scholes (BS) model, highlighting the VG model's sophistication and superior predictive capabilities.

Despite some drawbacks associated with the Practitioner Black-Scholes (PBS) model, it remains a technically intricate model that outperforms other renowned models. Therefore, any model surpassing the performance of the PBS model is noteworthy and deserves recognition in financial literature. In this context, the Practitioner Variance Gamma (PVG) model emerges as the standout candidate. Our analysis encompasses an entire year's worth of data from the global financial crisis period (subprime crisis) in 2008-2009.

To conduct our analysis, we methodically organized the data on a daily basis for 51 weeks and generated graphical representations of the RMSE of considered models for both in-sample and out-sample datasets. The calibration of the data involved the utilization of two MATLAB functions, namely lsqnonlin and fmincon.

Our empirical analysis conclusively demonstrates that the PVG model excels in the in-sample dataset. It consistently outperforms the PBS model in this context, highlighting its superior predictive accuracy and effectiveness. Furthermore, it is noteworthy that, in terms of delivering the best percentage distribution of results, the fmincon optimization technique consistently outperforms lsqnonlin, adding an additional layer of insight into the modeling process.

However, it is worth highlighting a surprising outcome from our analysis - the VG model performs exceptionally well in the forecast, surpassing both PVG and PBS. This unexpected result underscores the dynamic and nuanced nature of financial markets, where models may exhibit varying degrees of performance under different conditions.

In summary, this project's pioneering introduction of the practitioner's VG model into financial literature sheds light on the model's efficiency and its ability to outperform traditional approaches, even in challenging market conditions. The comprehensive analysis conducted using real-world data from the global financial crisis period reinforces the significance of our findings. Additionally, our exploration of optimization techniques provides valuable insights into enhancing model performance. Lastly, the unexpected outperformance of the VG model in the out-sample dataset underscores the need for ongoing research and adaptability in financial modeling to account for evolving market dynamics. The VG process surpasses both BS and PBS due to the diverse time frames associated with all levels of maturity.

Appendix

1. Codes Used For Black and Scholes:

Function for Black and Scholes Call Option Price Calculation:

```
function c = blackScholesPrice(s0, K, r, sigma, T)
    T=T./365;
    d1 = (log(s0./ K) + (r + sigma.^2 ./ 2) .* T) ./ (sigma .*
sqrt(T));
    d2 = d1 - sigma .* sqrt(T);
    c = s0.* normcdf(d1) - K.*exp(-r .* T).*normcdf(d2);
end
```

'lsqnonlin' Optimizer Function for Calibration:

```
function error = calcErr_BS(sigma, s0,K,r,T, market_price)

    optionPrices = blackScholesPrice(s0, K, r, sigma, T);
    error = optionPrices - market_price;
end
```

Script File for Calibrating using BS method (lsqnonlin):

```
clear sigma;
sigma = .1;

initialParams = sigma;
lowerBounds = eps;
upperBounds = 1;

fprintf('    Sigma\n=====================================\n')
disp([initialParams;lowerBounds;upperBounds])
options = optimoptions('lsqnonlin', 'Display', 'off');

% Calibration
sigma = lsqnonlin(@(sigma) calcErr_BS(sigma, s0,K,r,T,
market_price), ...
    initialParams, lowerBounds, upperBounds, [], [], [], [],
[],options);
disp('Estimated parameters using lsqnonlin:');
disp(sigma);
```

```
% Wednesday RMSE Check
c= blackScholesPrice(s0, K, r, sigma, T);
fprintf("(lsqnonlin)RMSE for BS Wednesday:
%f\n",rmse(market_price,c))

% Thursday RMSE Check
c_thur = blackScholesPrice(s0_thur, K_thur, r_thur, sigma,
T_thur);
fprintf("(lsqnonlin)RMSE for BS Thursday:
%f\n",rmse(market_price_thur,c_thur))
```

‘fmincon’ Optimizer Function for Calibration:

```
function error = calcErr_BS(sigma, s0,K,r,T, market_price)

    optionPrices = blackScholesPrice(s0, K, r, sigma, T);
    error = sum((optionPrices - market_price).^2);
end
```

Script File for Calibrating using BS method (fmincon):

```
clear sigma;

sigma = .1;

initialParams = sigma;
lowerBounds = eps;
upperBounds = 1;

fprintf('    Sigma\n====================================\n')
disp([initialParams;lowerBounds;upperBounds])
options = optimoptions('fmincon', 'Display', 'off');
% ----------------------Calibration----------------------
sigma = fmincon(@(sigma) calcErr_BS(sigma, s0,K,r,T,
market_price), ...
    initialParams,[], [], [],
[],lowerBounds,upperBounds,[],options);
disp('Estimated parameters using fmincon:');
disp(sigma);
% Wednesday RMSE Check
c= blackScholesPrice(s0, K, r, sigma, T);
fprintf("(fmincon)RMSE for BS Wednesday:
%f\n",sqrt(mean((market_price - c).^2)))
% Thursday RMSE Check
c_thur = blackScholesPrice(s0_thur, K_thur, r_thur, sigma,
T_thur);
```

```
fprintf("(fmincon)RMSE for BS Thursday:
%f\n",sqrt(mean((market_price_thur - c_thur).^2)))
```

2. Codes Used For Variance Gamma Process:

Function for Variance Gamma Call Option Price Calculation:

```
function c = VGcall_main_thur(s0,K,r,T,theta, sigma,nue)
T=T./365;
alpha =(-theta./sigma);
a=(alpha+sigma).^2;

num = 1 - (nue.*a)./2;
den = 1-(nue.* alpha.^2)./2;

d1 = real(log(s0./K)./(sigma.*sqrt(T)) +((r+
log(num./den).*(nue.^-1))./sigma +alpha +sigma).*sqrt(T));
d2 = d1 - sigma.*sqrt(T);

c = real(s0.*exp(a.*T./2).*(num.^(T./nue)).* normcdf(d1) -
K.*exp(-r.*T+(alpha.^2).*T.*.5).*(den.^(T./nue).*
normcdf(d2)));
end
```

‘lsqnonlin’ Optimizer Function for Calibration:

```
function error = calcErr_VGcall(params, s0,K,r,T,
market_price)
    theta= params(1); sigma = params(2); nue = params(3);

    optionPrices = VGcall_main(s0,K,r,T, theta,sigma, nue);
    error = optionPrices - market_price;
end
```

Script File for Calibrating using VG method (lsqnonlin):

```
clear theta sigma nue;
theta=.1;
sigma=.1;
nue=.1;

initialParams = [theta sigma nue];
lowerBounds = [-Inf,eps,eps];
upperBounds = [Inf,1,1];
fprintf('    Theta       Sigme
```

```
Nue\n====================================\n')
disp([initialParams;lowerBounds;upperBounds])
options = optimoptions('lsqnonlin', 'Display', 'off');
% Calibration
estimatedParams = lsqnonlin(@(params)
calcErr_VGcall(params,s0,K,r,T,market_price), ...
    initialParams, lowerBounds, upperBounds, [], [], [],
[],[],options);
disp('Estimated parameters using lsqnonlin:');
disp(estimatedParams);

% Wednesday RMSE Check
theta = estimatedParams(1);
sigma = estimatedParams(2);
nue= estimatedParams(3);
c= VGcall_main(s0,K,r,T,theta,sigma,nue);
fprintf("(lsqnonlin)RMSE for main VGcall Wednesday:
%f\n",rmse(market_price,c))

% Thursday RMSE Check
c_thur = VGcall_main_thur(s0_thur,K_thur,r_thur,T_thur,theta,
sigma,nue);
fprintf("(lsqnonlin)RMSE for main VGcall Thursday:
%f\n",rmse(market_price_thur,c_thur))
```

‘fmincon’ Optimizer Function for Calibration:

```
function error = calcErr_VGcall(params, s0,K,r,T,
market_price)
    theta= params(1); sigma = params(2); nue = params(3);

    optionPrices = VGcall_main(s0,K,r,T, theta,sigma, nue);
    error = sum((optionPrices - market_price).^2);
end
```

Script File for Calibrating using BS method (fmincon):

```
clear theta sigma nue;
theta=.1;
sigma=.1;
nue=.1;

initialParams = [theta sigma nue];
lowerBounds = [-Inf,eps,eps];
upperBounds = [Inf,1,1];
fprintf('    Theta       Sigme
```

```
Nue\n====================================\n')
disp([initialParams;lowerBounds;upperBounds])
options = optimoptions('fmincon', 'Display', 'off');
% ----------------------Calibration----------------------
estimatedParams = fmincon(@(params)
calcErr_VGcall(params,s0,K,r,T,market_price), ...
    initialParams, [], [], [], [], lowerBounds,
upperBounds,[],options);
disp('Estimated parameters using fmincon:');
disp(estimatedParams);

% Wednesday RMSE Check
theta = estimatedParams(1);
sigma = estimatedParams(2);
nue= estimatedParams(3);
c= VGcall_main(s0,K,r,T,theta,sigma,nue);
fprintf("(fmincon)RMSE for main VGcall Wednesday:
%f\n",sqrt(mean((market_price - c).^2)))

% Thursday RMSE Check
c_thur = VGcall_main_thur(s0_thur,K_thur,r_thur,T_thur,theta,
sigma,nue);
fprintf("(fmincon)RMSE for main VGcall Thursday:
%f\n",sqrt(mean((market_price_thur - c_thur).^2)))
```

3. Codes Used For Practitioner Black and Scholes Method:

Calculate Regress Coefficients for Practitioners Method:

```
function regress_coeffs = prac_regress_coeffs(impV,K,T)
n = length(impV);
RHS = ones(n,6);
for cnt =1:n
    RHS(cnt,1) = 1;
    RHS(cnt,2) = K(cnt);
    RHS(cnt,3) = K(cnt)^2;
    RHS(cnt,4) = T(cnt)/365;
    RHS(cnt,5) = (T(cnt)/365)^2;
    RHS(cnt,6) = K(cnt)*T(cnt)/365;
end
regress_coeffs = regress(impV,RHS);
end
```

Calculate Practitioners Volatility for Practitioners Method:

```
function [output] = prac_vol(p,K,T)
n = length(K);
output = p(1) +p(2).*K +p(3).*K.^2 +p(4).*(T./365)
+p(5).*(T./365).^2 +p(6).*K.*(T./365);
for i=1:n
    if output(i) <0
        output(i) = 0.01;
    end
end
end
```

Script File for Calibrating using Practitioners BS method:

```
regress_coeffs = prac_regress_coeffs(impV,K,T);

% Wednesday
prac_impV = prac_vol(regress_coeffs,K,T);
c = blackScholesPrice(s0, K, r, prac_impV, T);
fprintf("RMSE for PBS(Wednesday): %f\n",rmse(market_price,c))

% Thursday
prac_impV_thur = prac_vol(regress_coeffs,K_thur,T_thur);
c_thur = blackScholesPrice(s0_thur, K_thur, r_thur,
prac_impV_thur, T_thur);
fprintf("RMSE for PBS(Thursday):
%f\n",rmse(market_price_thur,c_thur))
```

4. Codes Used For Practitioner Variance Gamma Method:

Variance Gamma call price calculation function lsqnonlin:

```
function [c,sigma] =
VGcall_sigma_thur(s0,K,r,T,theta,nue,sigma_prac,sigma_bounds)
initialParams = sigma_bounds(1);
lowerBounds = sigma_bounds(2);
upperBounds = sigma_bounds(3);

options = optimoptions('lsqnonlin', 'Display', 'off');
for i =1:length(sigma_prac)
    sigma(i,1) = lsqnonlin(@(sigma) sigma_prac(i,1).^2 -
```

```
(nue.*theta.^2 + sigma.^2), ...

initialParams,lowerBounds,upperBounds,[],[],[],[],[],options);
end

T=T./365;
alpha =(-theta./sigma);
a=(alpha+sigma).^2;

num = 1 - (nue.*a)./2;
den = 1-(nue.* alpha.^2)./2;

d1 = log(s0./K)./(sigma.*sqrt(T)) +((r+ log(num./den).*(nue.^-
1))./sigma +alpha +sigma).*sqrt(T);
d2 = d1 - sigma.*sqrt(T);

c = s0.*exp(a.*T./2).*(num.^(T./nue)).* normcdf(d1) - K.*exp(-
r.*T+(alpha.^2).*T.*.5).*(den.^(T./nue).* normcdf(d2));
end
```

‘lsqnonlin’ Optimizer Function for Calibration:

```
function error = calcErr_sigma(params,
s0,K,r,T,sigma_prac,sigma_bounds, market_price)
    theta = params(1); nue = params(2);

    optionPrices = VGcall_sigma(s0,K,r,T, theta, nue,
sigma_prac,sigma_bounds);
    error = optionPrices - market_price;
end
```

Script File for Calibrating using Practitioners VG method (lsqnonlin):

```
warning('off', 'all');

theta = .01;
nue = .01;

initialParams = [theta, nue];
lowerBounds = [-1, eps];
upperBounds = [1, 1];

sigma_bounds = [.1,eps,1];

fprintf('     Theta         Nue
```

```
Sigma\n=====================================\n')
disp([initialParams,sigma_bounds(1);lowerBounds,sigma_bounds()
upperBounds,sigma_bounds(3)])
options = optimoptions('lsqnonlin', 'Display', 'off');

% Calibration
regress_coeffs = prac_regress_coeffs(impV,K,T);
sigma_prac = prac_vol(regress_coeffs,K,T);
estimatedParams = lsqnonlin(@(params)
calcErr_sigma(params,s0,K,r,T,sigma_prac,sigma_bounds,market_p
rice), ...
    initialParams,lowerBounds, upperBounds,[], [], [],
[],[],options);

disp('Estimated parameters using lsqnonlin:');
disp(estimatedParams);

% Wednesday RMSE Check
theta = estimatedParams(1);
nue = estimatedParams(2);
[c,sigma]=
VGcall_sigma(s0,K,r,T,theta,nue,sigma_prac,sigma_bounds);
% display(transpose(sigma))
fprintf("(lsqnonlin)RMSE for Sigma Wednesday:
%f\n",rmse(market_price,c))

% Thursday RMSE Check
sigma_prac_thur = prac_vol(regress_coeffs,K_thur,T_thur);
[c_thur,sigma_thur]=
VGcall_sigma_thur(s0_thur,K_thur,r_thur,T_thur,theta,nue,sigma
_prac_thur,sigma_bounds);
% display(transpose(sigma_thur))
fprintf("(lsqnonlin)RMSE for Sigma Thursday:
%f\n",rmse(market_price_thur,c_thur))
```

Function to Calculate Variance Gamma Call Option Price using fmincon:

```
function [c,sigma] =
VGcall_sigma(s0,K,r,T,theta,nue,sigma_prac,sigma_bounds)
initialParams = sigma_bounds(1);
lowerBounds = sigma_bounds(2);
upperBounds = sigma_bounds(3);


options = optimoptions('fmincon', 'Display', 'off');
for i =1:length(sigma_prac)
    sigma(i,1) = fmincon(@(sigma) sum((sigma_prac(i,1).^2 -
```

```
(nue.*theta.^2 + sigma.^2)).^2), ...

initialParams,[],[],[],[],lowerBounds,upperBounds,[],options);
end

T=T./365;
alpha =(-theta./sigma);
a=(alpha+sigma).^2;

num = 1 - (nue.*a)./2;
den = 1-(nue.* alpha.^2)./2;

d1 = log(s0./K)./(sigma.*sqrt(T)) +((r+ log(num./den).*(nue.^-
1))./sigma +alpha +sigma).*sqrt(T);
d2 = d1 - sigma.*sqrt(T);

c = s0.*exp(a.*T./2).*(num.^(T./nue)).* normcdf(d1) - K.*exp(-
r.*T+(alpha.^2).*T.*.5).*(den.^(T./nue).* normcdf(d2));
end
```

‘fmincon’ Optimizer Function for Calibration:

```
function error = calcErr_sigma(params,
s0,K,r,T,sigma_prac,sigma_bounds, market_price)
    theta = params(1); nue = params(2);

    optionPrices = VGcall_sigma(s0,K,r,T, theta, nue,
sigma_prac,sigma_bounds);
    error = sum((optionPrices - market_price).^2);
end
```

Script File for Calibrating using Practitioners VG method (fmincon):

```
warning('off', 'all');

theta = .1;
nue = .1;

initialParams = [theta, nue];
lowerBounds = [-1, eps];
upperBounds = [1, 1];

sigma_bounds = [.1,eps,1];

fprintf('    Theta        Nue
Sigma\n====================================\n')
```

```
disp([initialParams,sigma_bounds(1);lowerBounds,sigma_bounds(2
);upperBounds,sigma_bounds(3)])
options = optimoptions('fmincon', 'Display', 'off');
% ----------------------Calibration----------------------
regress_coeffs = prac_regress_coeffs(impV,K,T);
sigma_prac = prac_vol(regress_coeffs,K,T);
estimatedParams = fmincon(@(params)
calcErr_sigma(params,s0,K,r,T,sigma_prac,sigma_bounds,market_p
rice), ...
    initialParams,[], [], [], [],lowerBounds,
upperBounds,[],options);

disp('Estimated parameters using fmincon:');
disp(estimatedParams);
% Wednesday RMSE Check
theta = estimatedParams(1);
nue = estimatedParams(2);
[c,sigma]=
VGcall_sigma(s0,K,r,T,theta,nue,sigma_prac,sigma_bounds);
display(transpose(sigma))
fprintf("(fmincon)RMSE for Sigma Wednesday:
%f\n",sqrt(mean((market_price - c).^2)))

% Thursday RMSE Check
sigma_prac_thur = prac_vol(regress_coeffs,K_thur,T_thur);
[c_thur,sigma_thur]=
VGcall_sigma_thur(s0_thur,K_thur,r_thur,T_thur,theta,nue,sigma
_prac_thur,sigma_bounds);
display(transpose(sigma_thur))
fprintf("(fmincon)RMSE for Sigma Thursday:
%f\n",sqrt(mean((market_price_thur - c_thur).^2)))
```

References

1. Fiorani, Filo, “Option pricing under the Variance Gamma Process”, April 2004
2. Dilip B. Madan and Eugene Seneta “The Variance Gamma (V.G.) Model for Share Market Return”, The Journal of Business, vol 63. no.4, 1990, pages 511-524.
3. Dilip B Madan, Peter P. Carr and Eric C. Chang, “The Variance Gamma Process and Option Pricing”, European Finance Review, 2, 1998, pages 83.
4. Fisher Black and Myron Scholes, “The Pricing of Options and Corporate Liabilities”, Journal of Political Economy, Vol. 81, No. 3, May/June 1973, pages 637-654.
5. Ubbo F. Wiersema, “Brownian Motion Calculus”, John Wiley & Sons, Ltd. 2008, Edition 2020-2021
6. Fabrice Douglas Rouah, Gregory Vainberg, “Option Pricing Models and Volatility Using Execl® - VBA”, John Wiley & Sons, Ltd. 2007
7. Peter Carr, Hélyette Geman, Dilip B. Madan, Marc Yor, “The Fine Structure of Asset Returns: An Empirical Investigation*”, The Journal of Business , Vol. 75, No. 2 (April 2002), pp. 305-332,The University of Chicago Press, https://www.jstor.org/stable/10.1086/338705
8. Correcting the Bias in the Practitioner Black-Scholes Method, Yun Yin & Peter G. Moffatt, J. Risk Financial Manag. 2019