New York University

MSc in Financial Engineering

Fourier Transform and Heston Model in Option Pricing

Cristobal Gonzalez - Deniz Kural - Liam Trodden

Professor Daniel Totouom

12 December, 2019

Contents

1 Introduction **3**
1.1 Motivation . . . 3
1.2 Data . . . 3
1.3 Parameters . . . 3

2 Heston Model **5**
2.1 Description . . . 5
2.2 Calculation of Heston Model . . . 6
2.3 Closed Form of Heston Model . . . 6
2.4 Numerical Integration Methods . . . 7
2.4.1 Simpson's Rule . . . 7
2.5 Calibration . . . 8
2.5.1 Calibration Method Theory . . . 8
2.5.2 Calibration Results . . . 9

3 Fast Fourier Transform **10**
3.1 Description . . . 10
3.2 Calibration . . . 11

4 Analysis and Results **12**
4.1 Models Effectiveness . . . 12
4.2 Models Efficiency . . . 16
4.3 Comparison of Calibration Methods . . . 17
4.3.1 Potential Improvements to Calibration Method . . . 20

5 Conclusion **21**

6 References **22**

1 Introduction

1.1 Motivation

While the most notable contribution to the history of quantitative option pricing certainly comes from Black and Scholes in 1973, researchers soon noticed that prices found with the Black-Scholes method tend to differ from market prices. This would eventually lead to stochastic volatility models. In this work will look at that of Steve Heston (1993), an important model that would model volatility as stochastic and time-dependent. This model was able to price option in closed form, but doing so computationally proved difficult. Another major contribution to option pricing then came with Carr and Madan in 1999 with their version of the Fast Fourier transform, which was able to implement the dynamics of the Heston model for greatly improved efficiency.

This paper intends to analyze and compare European call option pricing using both the closed form Heston model and the Fast Fourier Transform method (Carr-Madan). We intend to determine when it is appropriate to use FFT methods for pricing European calls based on the efficiency of its use with a variety of calibration techniques.

To test the Carr-Madan FFT, we implement a model in Python for the closed form Heston method. Then, we calibrate the parameters based on market data using two calibration methods: one that minimizes squared differences in price between the model and the market, and another that does the same with option implied volatility.

Next, we implement a model in Python based on the Carr-Madan FFT, calling on the characteristic function as defined by Heston stock dynamics. After calibrating this model similarly to with the closed form model, we compare the accuracy and computational efficiency of both models. While it is known that the FFT model is fast and flexible, we intend to use these results to investigate the potential errors or shortcomings of the FFT and pinpoint where and why those might arise.

1.2 Data

The data used in this paper are S&P 500 index options from January to December of 2018. Underlying values, interest rates, and the option information (prices, maturities, strikes, implied volatility, and vega) were downloaded from the Wharton Research Data Services. The prices are the lowest closing Ask across all the exchanges available in the platform.

To make the analysis more simple, we took call options with three months of maturity, in the window of 89 to 94 days. Therefore the models' performances will be showed against only different strike prices. Since the data is obtained from many different exchanges, there is information for a wide variety of strikes with competitive prices. There are 98 days of option information and around 91 strikes per day, being a total of almost 9000 call options.

1.3 Parameters

Before using the model with data, we test the sensitivity of both models to changes in rho and volatility of the volatility. For the other input parameters, there are general principles that can serve as guidance to choose initial values before calibrating. Choosing values for these two parameters can be more obscure. The below plots visualize this analysis.

First for the Heston Closed Form:

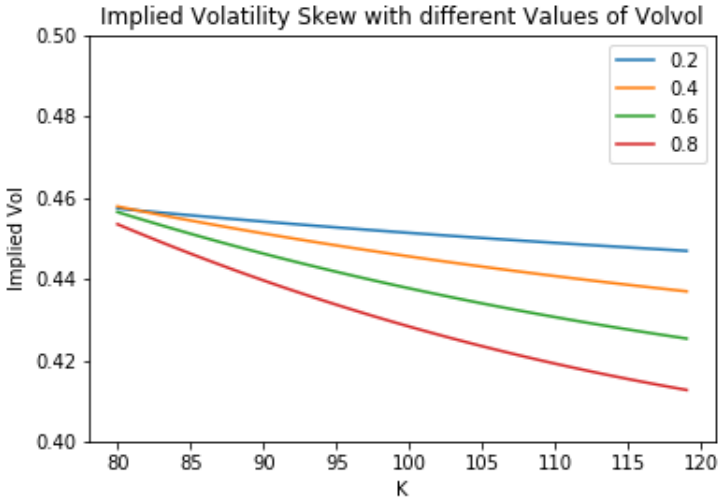

Figure 1.1: Closed Form Sensitivity to sigma

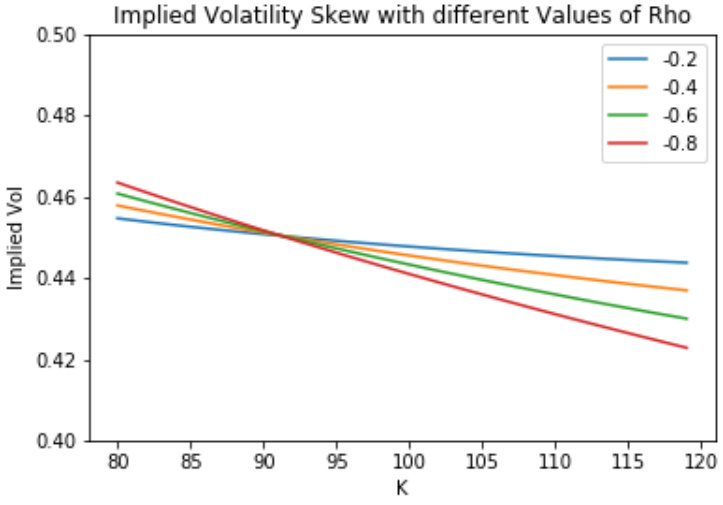

Figure 1.2: Closed Form Sensitivity to Rho

And for the Fast Fourier Transform:

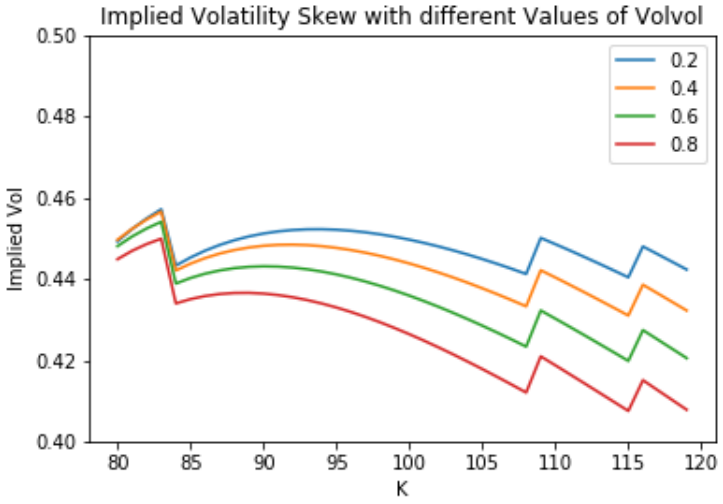

Figure 1.3: Fast Fourier Sensitivity to Sigma

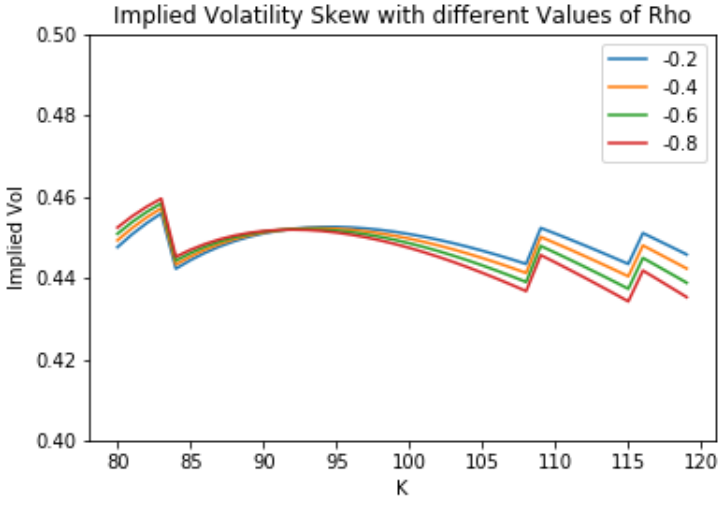

Figure 1.4: Fast Fourier Sensitivity to Rho

2 Heston Model

2.1 Description

Heston model was developed by Professor Steve Heston in 1993. The model is based on a stochastic volatility assumption, and is used primarily for pricing European options. In contrast to the Black-Scholes model, Heston's model does not hold volatility constant. Instead, it assumes volatility is arbitrary. One of the special characteristics of this model is that it provides closed-form solution

for European options. Thus, when there is a known correlation between a stock's price and its volatility, the solution can be obtained.

2.2 Calculation of Heston Model

The model can be shown as following the dynamics:

$$dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t \tag{2.1}$$

$$dV_t = \kappa(\theta - V_t)dt + \sigma\sqrt{V_t} dZ_t \tag{2.2}$$

$$dW_t dZ_t = \rho dt \tag{2.3}$$

The parameters in Equations (2.1),(2.2) and (2.3) are clarified below,
μ is the drift coefficient of the stock price,
σ is the volatility of the variance diffusion,
θ is the long-term mean of variance,
κ is the rate of mean reversion,
ρ is the correlation between the Brownian motions,
S_t is the price process,
V_t is the variance process.

Equation (2.3) indicates leverage effect, which shows that returns and implied volatility are negatively correlated. To create this model, Heston (1993) assumed a log-normal distribution for the underlying stock and Cox-Ingersoll-Ross process (CIR process) (1985) for V_t [1]. As mentioned, this model has some advantages over Black-Scholes Model. Volatility affects can not be ignored in financial markets for the duration of the option's life and Heston model takes this into account where as Black-Scholes does not. Moreover, Heston models volatility as a mean reverting process which is observed in financial markets [2]. Lastly, he shows correlation between underlying asset returns and volatility which is also observed in financial market movements.

2.3 Closed Form of Heston Model

Heston (1993) assumed the solution form would be similar to the Black-Scholes model. The formula for present value of European call option can be seen at Equation (2.4) below:

$$C_0 = S_0 e^{-qT} P_1 - K e^{-rT} P_2 \tag{2.4}$$

So, to calculate call value, both P_1 and P_2 must be found. Thus, call value in Heston model can be calculated by finding these values with the usage of Equations (2.1), (2.2), (2.3) and substituting these values inside the Equation (2.4).

The give probabilities for P_1 and P_2 can be found via Fourier Inversion Transformation,

$$P_j = \frac{1}{2} + \frac{1}{\pi}\int_{\phi=0}^{+\infty} Re[\frac{e^{-i\phi lnK} f_j(\phi|x_0, \nu_0, T)}{i\phi}]d\phi \tag{2.5}$$

Heston (1993) uses the characteristic function as in Equation (2.6),

$$f_j(\phi|x_0, \nu_0, T) = exp[C_j(\phi|T) + D_j(\phi|T)\nu_0 + i\phi x_0] \tag{2.6}$$

where the parameters of characteristic function can be seen in the below equations,

$$C_j(\phi|T) = i\phi(r-q)T + \frac{a}{\xi^2}[(b_j - i\phi\rho\xi + d_j)T - 2ln\frac{1-g_j e^{d_j T}}{1-g_j}] \tag{2.7}$$

$$D_j(\phi|T) = [\frac{b_j - i\phi\rho\xi + d_j}{\xi^2}][\frac{1-e^{d_j T}}{1-g_j e^{d_j T}}] \tag{2.8}$$

$$g_j(\phi) = \frac{b_j - i\phi\rho\xi + d_j}{b_j - i\phi\rho\xi - d_j} \tag{2.9}$$

$$d_j(\phi) = \sqrt{(i\phi\rho\xi - b_j)^2 - \xi^2(2i\phi u_j - \phi^2)} \tag{2.10}$$

$$u_1 = \frac{1}{2} \tag{2.11}$$

$$u_2 = -\frac{1}{2} \tag{2.12}$$

$$a = \theta\omega \tag{2.13}$$

$$b_1 = \theta + \psi - \rho\xi \tag{2.14}$$

$$b_2 = \theta + \psi \tag{2.15}$$

2.4 Numerical Integration Methods

2.4.1 Simpson's Rule

There are different methods to solve Equation (2.5) as well, including Simpson's rule. The logic behind the Simpson's rule is that the function for numerical integration is divided into two equal parts. As can be seen in the below Equation (2.16), the partition points are specified as a, a+h, and a+2h [3].

$$\int_a^{a+2h} f(x)dx \approx \frac{h}{3}[f(a) + 4f(a+h) + f(a+2h)] \tag{2.16}$$

The error of this numerical integration method can be also found and can be seen from the Equation (2.17) and (2.18),

$$f(a+h) = f(a) + hf'(a) + \frac{1}{2!}h^2 f''(a) + \frac{1}{3!}h^3 f'''(a) + ... \tag{2.17}$$

$$f(a+2h) = f(a) + 2hf'(a) + 2h^2 f''(a) + \frac{4}{3}h^3 f'''(a) + ... \tag{2.18}$$

With defining $F(x) = \int_a^x f(t)dt$ and using Taylor series definition, equation can be modified as,

$$F(a+2h) = F(a) + 2hF(a) + 2h^2 F''(a) + \frac{4}{3}h^3 F'''(a) + \frac{2}{3}h^4 F^{(4)} + ... \tag{2.19}$$

Then, using $F^{(n+1)}(a) = f^{(n)}(a)$,

$$F(a+2h) = F(a) + 2hF(a) + 2h^2 F''(a) + \frac{4}{3}h^3 F'''(a) + \frac{2}{3}h^4 F^{(4)} + ... \tag{2.20}$$

Subtracting RHS of Equation (2.16) from Equation (2.20), error formula can be found as,

$$Error = -\frac{h^5}{90} f^{(4)}(\xi) \tag{2.21}$$

Thus, final form Simpson rule can be observed at Equation (2.22) with error formula in Equation (2.23),

$$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)] \tag{2.22}$$

$$Error = -\frac{1}{90}(\frac{b-a}{2})^5 f^{(4)}(\xi), \xi \in (a, a+2h) \tag{2.23}$$

In order to find the integral of Heston, Equation (2.5) should be calculated. The implementation of this has been done in Python.

2.5 Calibration

2.5.1 Calibration Method Theory

There are six parameters that need to be estimated in Heston model. These are $\kappa, \theta, \sigma, V_0, \rho \, and \, \lambda$. The estimation of these parameters are crucial since even small change in these values will pose a great impact on the correctness of the price estimation. A popular approach to find optimal values is to minimize the difference between the model and real market prices [3]. The minimization is solved as nonlinear least squares optimization problem. Thus, corresponding formula for least square method can be observed at Equation (2.24),

$$min_\Omega S(\Omega) = min_\Omega \sum_{i=1}^{N} [C_i^\Omega(K_i, T_i) - C_i^M(K_i, T_i)]^2 \tag{2.24}$$

where N is the number of options , Ω is a vector of parameter values, $C_i^\Omega(K_i, T_i)$ and $C_i^M(K_i, T_i)$ are prices the model and market prices for the i_{th} option, strike K_i and maturity T_i.

Another popular approach for minimization function can be observed at the Equation (2.25). In this case, the minimization is done based on implied volatility. So, the *vega* term can be seen in the denominator of Equation (2.25) as the *vega* of i_{th} option in market. This approach is used later, in 4.3

$$min_\Omega S(\Omega) = min_\Omega \sum_{i=1}^{N} \frac{[C_i^\Omega(K_i, T_i) - C_i^M(K_i, T_i)]^2}{vega_i^M} \tag{2.25}$$

For the calibration of model, sequential least squares method will be used. This is selected since sequential quadratic programming is a crucial tool for solving non-linear optimization problems in the real world. General problem in sequential quadratic method can be observed below at Equation (2.26) [5]:

$$min f(x) \, such \, that \, h(x) = 0 \, and \, g(x) \leq 0 \tag{2.26}$$

where x indicates a vector and f(x), g(x) and h(x) are potentially nonlinear functions. Moreover, Lagrangian function for this problem can be defined as follows,

$$\mathcal{L}(x, \lambda, \sigma) = f(x) - \lambda^T(b(x) - (y_i)^2) - \sigma^T c(x) \tag{2.27}$$

where λ and σ are Lagrange multipliers. d(k) shows an incremental change to the objective function, so a minimization sub-problem can be obtained with d(k) and this problem is quadratic and thus

must be solved with non-linear methods such as sequential least squares method. The subproblem is shown in the following Equations from (2.28) to (2.31),

$$min_d \, such \, that \tag{2.28}$$

$$f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \sigma_k) d \tag{2.29}$$

$$b(x_k) + \nabla b(x_k)^T d \geq 0 \tag{2.30}$$

$$c(x_k) + \nabla c(x_k)^T d = 0 \tag{2.31}$$

Note that sequential least squares method is implemented in Python with *minimize(method='SLSQP')* function.

2.5.2 Calibration Results

To calibrate the Heston closed form model, we use the first method, that of minimizing squared difference in prices. Unlike the Fast Fourier model, which will be explained later, to calibrate the Heston closed form we give the model one set of inputs for one output. Since computations of this took so long, we used a total of 545 data points from the first month of data, coming from eight different days.

We ran it with the following initial parameters:

Initial Parameters	
Mean reversion κ	1.5
Correlation between Brownian Motions ρ	-0.4
Volatility of the volatility σ	0.6
Long-term mean of variance θ	0.03
Variance initial value V_0	0.014

Figure 2.1: Initial parameter for the calibration.

For better results, it is important to use somewhat realistic initial parameters. We took V_0 as 0.014, since this is equal to $(12\%)^2$, with 12% being a realistic volatility for an equity like the S&P 500. Similarly, we used a θ near that number. Since there is a historically observed negative correlation between risk and prices, we used a negative ρ.

Calibrated Parameters	
Mean reversion κ	0.0000
Correlation between Brownian Motions ρ	-0.6574
Volatility of the volatility σ	0.4303
Long-term mean of variance θ	0.0282
Variance initial value V_0	0.0144

Figure 2.2: Calibrated parameter.

From the calibrated model, we took an aggregate (mean) of the error terms:

Error Function = 11.8363

3 Fast Fourier Transform

3.1 Description

The idea of using Fast Fourier Transform (FFT) for pricing options comes from Carr and Madan's work, published in 1999 [4]. In their paper, they developed a new technique that uses FFT for valuing options more efficiently. Their technique uses the characteristic function of the risk-neutral density, similar to older techniques assume. Then, they develop an expression for the Fourier Transform of the option price. Finally, they use an algorithm of an inverse Fourier transform to solve the option price numerically.

The basic definition of Fourier Transform and inverse Fourier Transform can be seen below at Equation (3.1) and (3.2), correspondingly ,

$$\mathfrak{F}f(x) = \int_{-\infty}^{\infty} e^{i\phi x} f(x)dx = F(\phi) \tag{3.1}$$

$$\mathfrak{F}^{-1}f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\phi x} F(\phi)d\phi = f(x) \tag{3.2}$$

In these equations, f(x) represents the log return natural density function and $F(\phi)$ represents the characteristic function. Carr and Madan (1999) developed analytical expressions for the Fourier transform of an option price. So, as in the paper if we select k as the log of the strike price K and if we choose $C_T(k)$ as the desired value of a T maturity call option with strike $e^{(k)}$, the risk neutral density of log price s_T be $q_T(s)$, the characteristic function will be,

$$\phi_T(u) = \int_{-\infty}^{\infty} e^{ius} q_T(s)ds \tag{3.3}$$

Then, $C_t(k)$ can be written as,

$$C_T(k) = \int_{k}^{\infty} e^{-rT}(e^s - e^k)q_T(s)ds \tag{3.4}$$

At this point, since $C_t(k)$ is observed to be not square integrable, call price is modified to be square integrable as can be observed in Equation (3.5),

$$c_T(k) = e^{\alpha k} C_T(k) \, for \, \alpha > 0 \tag{3.5}$$

After this modification, $c_t(k)$ is now square integrable in k over real line. Thus, we can calculate Fourier and Inverse Fourier Transform of , $c_t(k)$ which can be seen at Equations (3.6) and (3.7),

$$F_{c_T}(\phi) = \int_{-\infty}^{\infty} e^{i\phi k} c_T(k)dk \tag{3.6}$$

$$c_T(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\phi k} F_{c_T}(\phi)d\phi \tag{3.7}$$

By substituting Inverse Fourier of $c_T(k)$ (Equation (3.7)) into Equation (3.5) , we get

$$C_T(k) = e^{-\alpha k} c_T(k) \tag{3.8}$$

$$= e^{-\alpha k}\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\phi k} F_{c_T}(\phi)d\phi \tag{3.9}$$

$$= e^{-\alpha k}\frac{1}{\pi}\int_0^\infty e^{-i\phi k}F_{c_T}(\phi)d\phi \tag{3.10}$$

Thus, $F_{c_T}(\phi)$ expression is determined as follows:

$$F_{c_T}(\phi) = \int_{-\infty}^{\infty} e^{\alpha k}e^{i\phi k}e^{-rT}\int_k^\infty (e^{x_T} - e^k f_T(x_T)dx_T dk \tag{3.11}$$

$$= \int_{-\infty}^{\infty} f_T(x_T)e^{-rT}\int_{-\infty}^{x_T}(e^{x_T+\alpha k} - e^{(\alpha+1)k})e^{i\phi k}dkdx_T \tag{3.12}$$

$$= \frac{e^{-rT}}{\alpha^2+\alpha-\phi^2+i(2\alpha+1)\phi}\int_{-\infty}^{\infty} e^{(\alpha+1+i\phi)}f_T(x_T)dx_T \tag{3.13}$$

$$= \frac{e^{-rT}}{\alpha^2+\alpha-\phi^2+i(2\alpha+1)\phi}\int_{-\infty}^{\infty} e^{(-\alpha i-i+\phi)ix_T}f_T(x_T)dx_T \tag{3.14}$$

$$= \frac{e^{-rT}F_{C_T}(\phi-(\alpha+1)i)}{\alpha^2+\alpha-\phi^2+i(2\alpha+1)\phi} \tag{3.15}$$

So, after finding $F_{c_T}(\phi)$, this function can be implemented in Equation(3.10) and FFT can be taken to determine call value of the option.

Carr and Madan also examined optimized α in their study. They have found that Equation (3.15) should be integrable, thus $F_{c_T}(0)$ should be finite. To accomplish this, $\Phi_T(-(\alpha+1)i)$ should be finite and also:

$$E[S_T^{\alpha+1}] < \infty \tag{3.16}$$

should be satisfied. So, one can determine α with the given restrictions or boundaries for the selection of α. Carr and Madan (1999) further extend Fourier Transform in out-of-money options since these options have no intrinsic value. However, this type of options are not considered in this article.

FFT in discrete form can be seen at the following equation:

$$w(k) = \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(k-1)}x(j)\, for\, k = 1, 2, ..., N \tag{3.17}$$

Where N is in the order of 2.With the utilization of this form, Carr and Madan formed the following expression,

$$C(k_u) = \frac{e^{-\alpha k_u}}{\pi}\sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)}e^{ibv_j}\kappa(v_j)\frac{\eta}{3}[3+(-1)^j-\delta_{j-1}] \tag{3.18}$$

This equation for call price is written under the assumption of Simpson's rule. Hence, Equation (3.18) in Carr Madan shows the application of FFT for option pricing. To price options with this formula, n and α values are needed to be selected properly. Moreover, note that the term corresponding to x(j) in Equation (3.17) at Equation (3.18) in Carr Madan returns a vector.

3.2 Calibration

For calibrating this model, we use the same data and initial parameters as the Heston Closed Form Calibration in section 2.5.

The Fast Fourier model works differently than the closed form model in that it does not have a strike input, and instead it returns a vector of prices at some strikes around the money. In this way,

the model calculates a number of prices at once, significantly reducing the number of iterations required to calibrate it.

Calibrated Parameters	
Mean reversion κ	153.9071
Correlation between Brownian Motions ρ	-0.7273
Volatility of the volatility σ	38.9586
Long-term mean of variance θ	0.0285
Variance initial value V_0	0.1772

Figure 3.1: Calibrated parameter.

After calibration, we received the following average squared difference:

Error Function = 46.3814

Notably, this is a higher aggregate error term than in the closed form. However, aggregates often fail to tell the whole story, and it may be the case that the Fast Fourier model matched many points perfectly but did extremely poorly at points very far from at the money. We will need more data to safely conclude that one model is more practically accurate than another.

4 Analysis and Results

4.1 Models Effectiveness

To test the accuracy of the models, we use a subset of the first quarter of data, including contracts from four different days throughout the quarter. We use the calibrated each model on this quarter of data, and use those parameters for each model to test how close the model fit to market prices and implied volatility on each of the test days.

Calibrated parameters for closed form:

Calibrated Parameters	
Mean reversion κ	0.7035
Correlation between Brownian Motions ρ	-0.6751
Volatility of the volatility σ	0.9728
Long-term mean of variance θ	0.2879
Variance initial value V_0	0.0000

Figure 4.1: Calibrated parameter.

Here is a plot of the first test day:

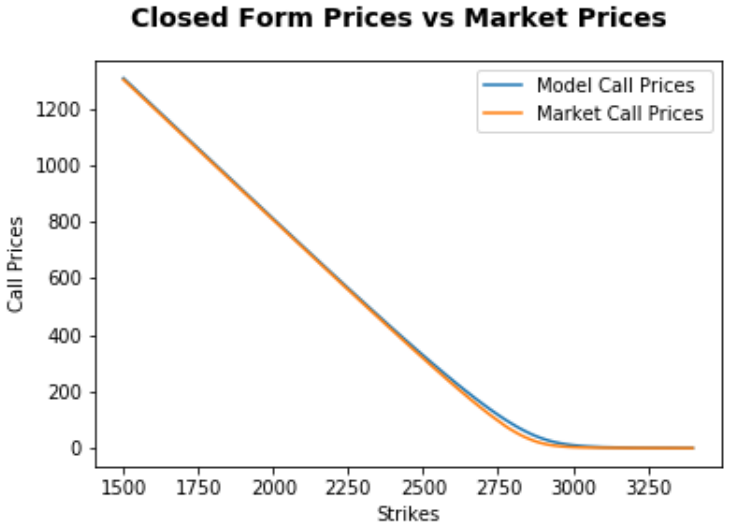

Figure 4.2: Closed Form Prices vs Market.

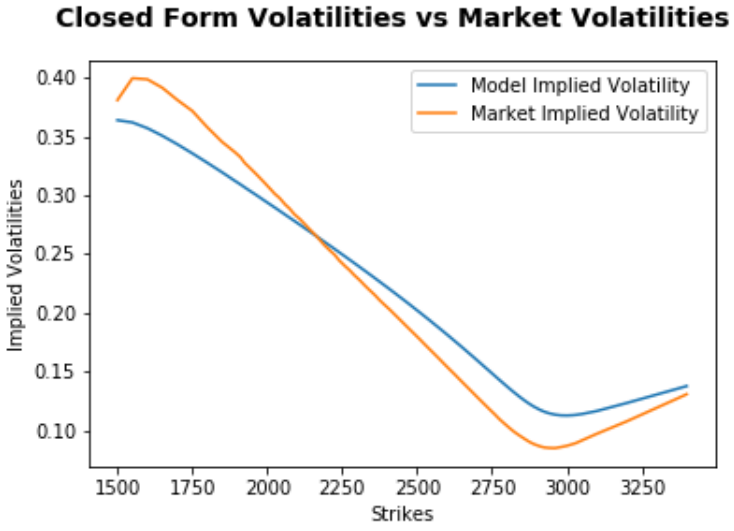

Figure 4.3: Closed Form Volatility vs Market.

The calibrated values for the Fast Fourier model for this data set:

Calibrated Parameters	
Mean reversion κ	125.5309
Correlation between Brownian Motions ρ	-0.7.103
Volatility of the volatility σ	29.9473
Long-term mean of variance θ	0.0440
Variance initial value V_0	0.0553

Figure 4.4: Calibrated parameter.

And the plots for the first test day for FFT:

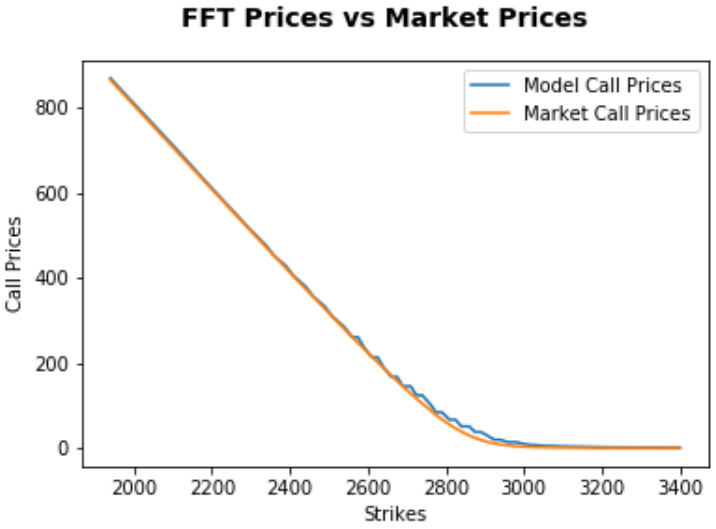

Figure 4.5: Fast Fourier Prices vs Market.

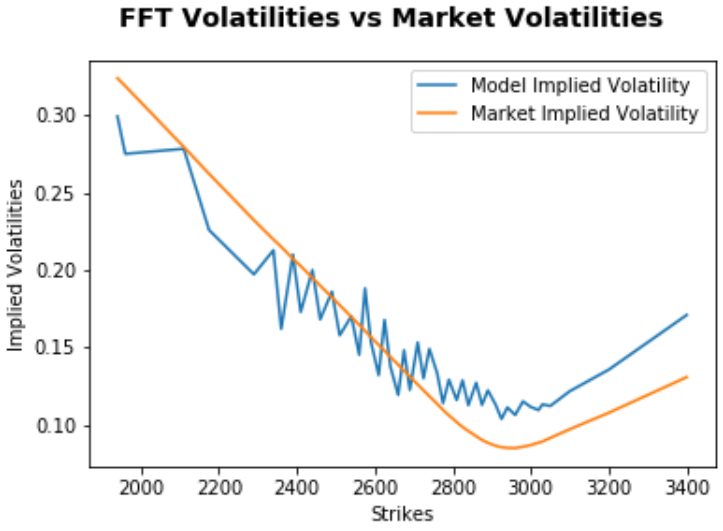

Figure 4.6: Fast Fourier Volatility vs Market.

On basic inspection, it seems that the prices given by these models are both relatively accurate, though some oscillation can be seen near the money in the Fast Fourier model. As for implied volatility, neither model fits very well to the market smile, with Fast Fourier showing some aggressive oscillation. To address the argument that these results may be too dependent on the chosen test dates, the following tables show some measurements on the entire first quarter of data. The data is separated into quartiles, with the first row showing the net difference between the model and the market, and the second row showing the average of absolute differences between the model and the market. Larger, more positive, values in the first row show an upward error in the model above the market overall, and larger values in the second row show inaccuracy of the model with respect to the market.

Closed Form Prices			
In-the-Money	ITM and ATM	ATM and OTM	Out-the-Money
0.9958	-0.4320	2.2959	1.2780
3.8268	8.0926	8.5197	1.9305

Figure 4.7: Aggregate level price error

Closed form Volatility			
In-the-Money	ITM and ATM	ATM and OTM	Out-the-Money
-0.0591	-0.0158	-0.00076	0.00008
0.0630	0.0261	0.0175	0.0133

Figure 4.8: Aggregate Level Volatility Error

A similar analysis can be done for the Fast Fourier model.

Fast Fourier Prices			
In-the-Money	ITM and ATM	ATM and OTM	Out-the-Money
-6.0292	-8.2438	-1.0719	1.7108
8.9698	12.5684	9.0875	1.9717

Figure 4.9: Aggregate level price error

Fast Fourier Volatility			
In-the-Money	ITM and ATM	ATM and OTM	Out-the-Money
-0.0663	-0.0297	-0.0072	0.0143
0.0698	0.0348	0.0196	0.0158

Figure 4.10: Aggregate Level Volatility Error

4.2 Models Efficiency

To test the computational efficiency of both of these models, we time the calibration process. Since calibration involves running the optimization function (in this case the call price functions) multiple times, any differences in efficiency are exacerbated by the calibration process. To compare, we run the calibration on a small amount of data for each model, and record the time to calculate with increasing amounts of data. We used increments of 85 rows of data, since we found in our data that this is close to the amount of options seen on the average day in our data set. The outcome is shown in the following plots:

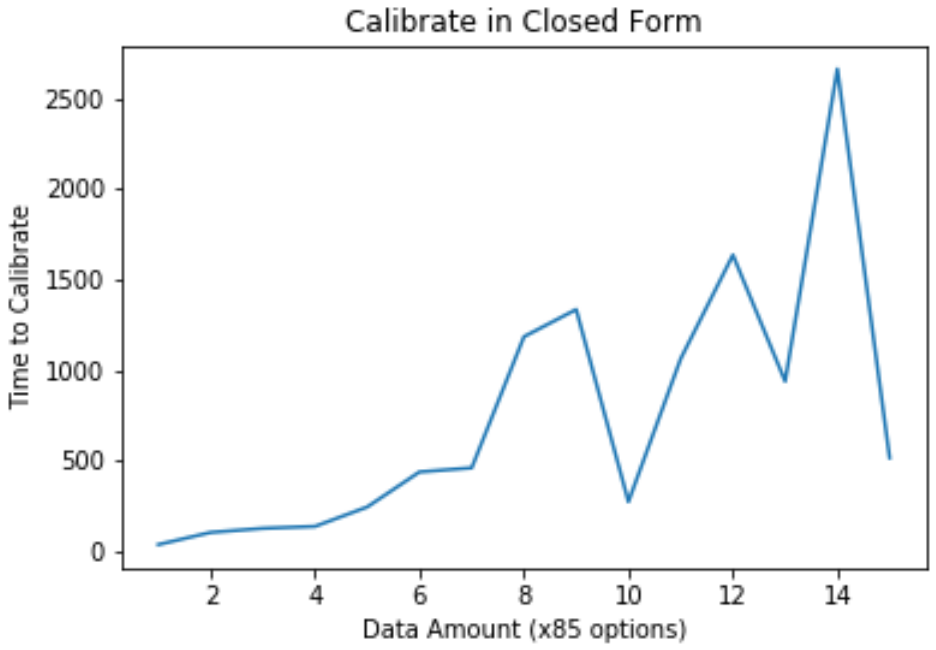

Figure 4.11: Time to Calibrate Closed Form

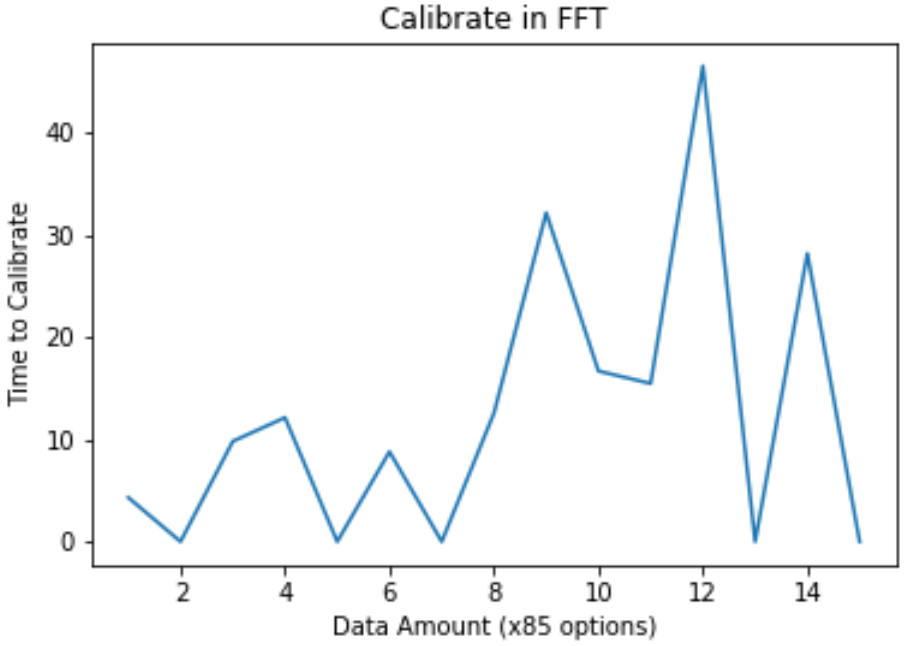

Figure 4.12: Time to Calibrate Fast Fourier

Based on this outcome, we are unable to conclude if the time to compute has a linear relationship with the amount of underlying data, or if the relationship is similar between the two models. Notice however, the scale of the two outcomes. Even for this small amount of data, calibrating the closed form Heston model took close to 20 minutes for some data amounts. This makes it hard to conceive that this could ever be a practical model for use with data sets that would often be much larger than we are using for this test. The Fast Fourier model, on the other hand, never took more than one minute to calibrate. This could likely be further optimized beyond what we were able to do with our model, but it is clear that for those who trade frequently, and would thus like to frequently calibrate their pricing model to the market, the Carr-Madan Fast Fourier model is a strong option.

4.3 Comparison of Calibration Methods

Up until now, we have calibrated using a minimization of squared differences in price method. However, we want to compare this with another potential calibration technique: that of minimizing the squared difference of implied volatility. We do this by dividing by the market vega of a given contract. Using this calibration method, we obtain the following optimized parameters:

Heston closed form:

Calibrated Parameters	
Mean reversion κ	1.5061
Correlation between Brownian Motions ρ	-0.6921
Volatility of the volatility σ	0.5364
Long-term mean of variance θ	0.1241
Variance initial value V_0	0.0000
Initial Guess	0.35
Vol Accuracy	0.00001

Figure 4.13: Calibrated parameter.

Calibrated Parameters	
Mean reversion κ	1.3062
Correlation between Brownian Motions ρ	-0.7263
Volatility of the volatility σ	0.7838
Long-term mean of variance θ	0.0000
Variance initial value V_0	0.0000
Initial Guess	0.35
Vol Accuracy	0.00001

Figure 4.14: Calibrated parameter.

The last two parameters, Initial Guess and Vol Accuracy, are necessary for using the Newton Raphson method to convert the prices of the model into implied volatility by iterating on the Black-Scholes formula.

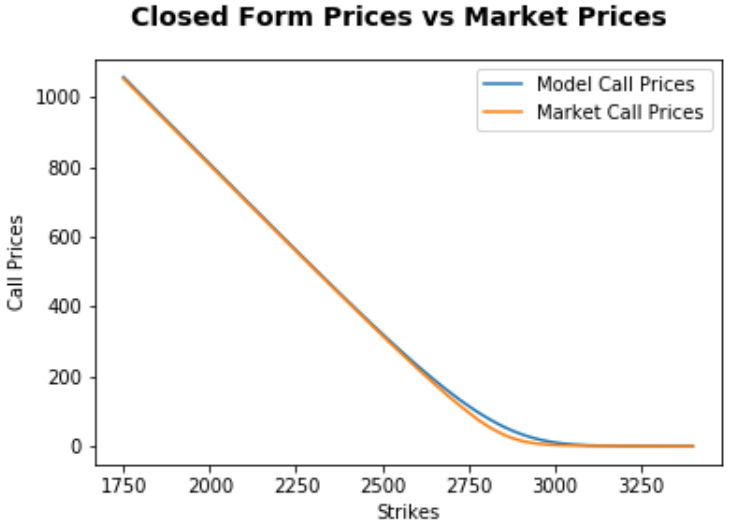

Figure 4.15: Closed Form Prices vs Market.

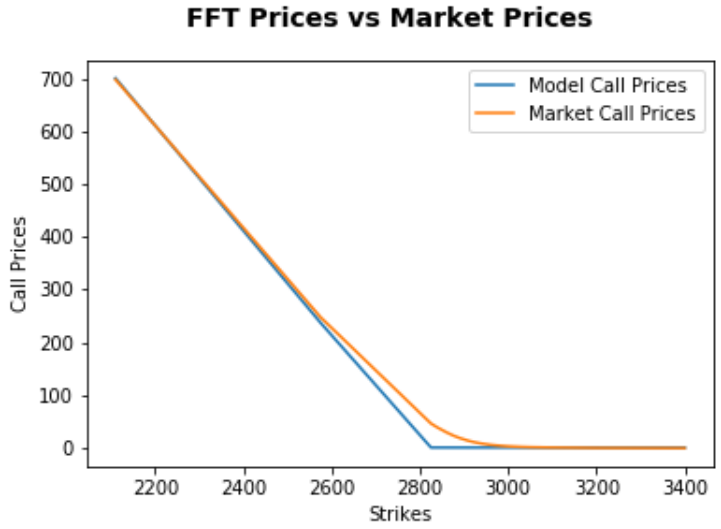

Figure 4.16: Fast Fourier Prices vs Market.

With basic visual inspection, it is clear that this is less accurate than the former calibration method, likely due to the inaccurate implied volatility smile obtained by the model, as seen earlier.

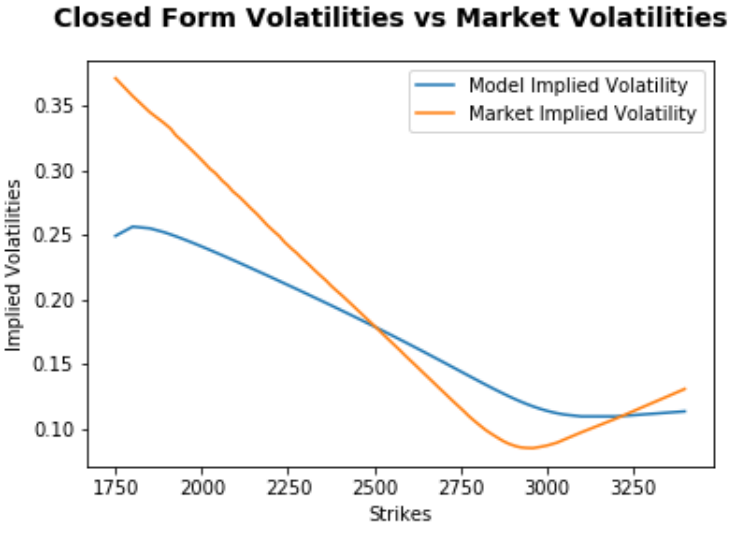

Figure 4.17: Closed Form Volatility vs Market.

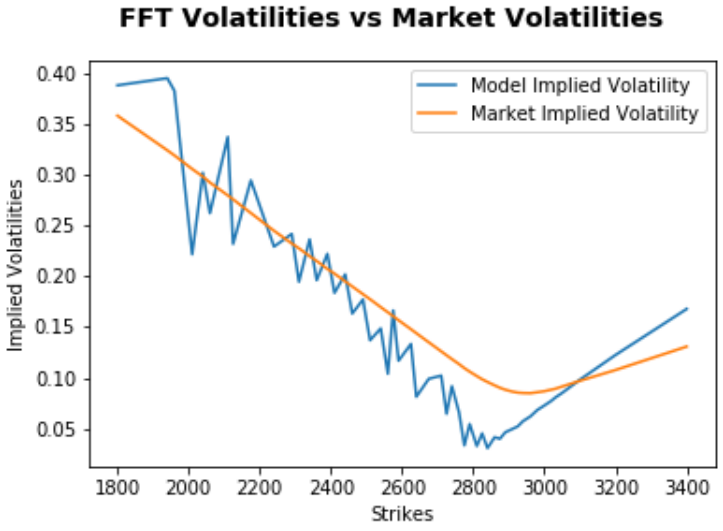

Figure 4.18: Fast Fourier Volatility vs Market.

4.3.1 Potential Improvements to Calibration Method

We realize there is room for improvement in the calibration method beyond simply the choice between optimizing prices or implied volatility. In this paper, constrained optimization was used, with constraints that all parameters be positive, except ρ, which must be between -1 and 1. Since constrained optimization can sometimes produce sub-optimal results, one potential remedy is to map each parameter to a term that is naturally bounded to the desired constraint. We attempted to do this by mapping ρ to a hyperbolic tangent function and the remaining functions to an exponential, as shown here:

$$\rho \to \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.1}$$

$$other\ parameters \to e^x \tag{4.2}$$

However, we were not successful with this attempt. Plots for the closed form calibration came back identical to the hard-constrained version, and Fast Fourier calibration failed with the optimizer reaching the maximum possible number of iterations without a successful output. This is an area where other researchers can attempt to improve on our calibration methods.

5 Conclusion

In this paper, we have outlined the derivations for the Heston stochastic volatility model and the Carr-Madan Fast Fourier Transform model. Hopefully, readers will understand the motivations behind the development of both models, as well as be able to build implementations themselves to replicate the results obtained here.

After testing both models, it is clear that the errors from the Fast Fourier transform method are in fact higher than those of the Heston closed form model. Conversely, the time it takes to calibrate the closed form model makes it not viable for any practical use, and the slightly more accurate results do not justify the order of magnitude difference in calibration time.

Further, we have explored the importance of calibration in using any model with practical data. For our case, we have shown that calibration via minimizing squared differences in prices is more accurate than that of implied volatility when obtaining call option prices.

6 References

[1]Yang, Yuan, "Valuing a European option with the Heston model" (2013). Thesis. Rochester Institute of Technology.

[2] Crisostomo, R. (2014, December). An Analysis of the Heston Stochastic Volatility Model: Implementation and Calibration using Matlab. Retrieved from https://arxiv.org/pdf/1502.02963.pdf

[3] Moodley, N. (2005) The Heston Model: A Practical Approach with Matlab Code. B.Sc. Thesis, University of the Witwatersrand, Johannesburg.

[4] Carr, P. and Madan, D. B. (1999), 'Option evaluation the using fast Fourier transform', *Journal of Computational Finance* 2(4), 61-73.

[5] Goodman, B. (2016). *Sequential quadratic programming.* Retrieved from https://optimization.mccormick.northwestern.edu/index.php/Sequential_quadratic_programming