UNIVERSITY OF CAPE TOWN

A survey and implementation of some calibration algorithms for the SABR and Heston models

Author: Caroline WEBBER *Supervisor:* Prof. Ronald BECKER

February 17, 2014

Submitted in partial fulfillment of the M.Phil degree in Mathematical Finance in the School of Economics.

Abstract

This thesis surveys and implements some calibration methods for the SABR and Heston models. Hagan (2002) examined the effect of the SABR parameters on the skew in order to determine which parameters may be redundant. Hagan and West (2005) found that by fixing one of the parameters in the SABR model, the remaining parameters were stable over time. We implement a SABR calibrator to confirm that the parameters are stable over time. We then examine the effects of the five Heston parameters on the skew in order to determine if any of the parameters are redundant. Calibrators where some parameters have been fixed and calibrators where no parameters have been fixed are implemented. The performance of these calibrators is then compared based on three criteria: the stability of the parameters over time, the fit of the solution and the computational efficiency of the calibrator. We find that the Heston parameters are more stable if the redundant parameters are fixed, the computation time is less and the fit is slightly worse. All implementations are done in the context of the South African market. The calibrators are programmed in Matlab and the code is included in the appendix.

Acknowledgements

I would like to thank my original supervisor, the late Graeme West, for the energy and time he put into starting me off on this thesis. You are greatly missed.

Thanks to Prof. Ronnie Becker for stepping in.

Thank you to all the BSM people who helped me through: Dylan, Moe, Cong, Odwa, Mark and Susan. Thank you to Lydia for helping me dig up some spreadsheets. Thank you Jo for proof-reading (twice!).

Plagiarism declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and to pretend that it is one's own.
2. Each significant contribution to, and quotation in, this report from the work of other people has been attributed, and has been cited and referenced.
3. This thesis is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature:

Contents

1 Introduction

Vanilla European options or European futures options are often priced and hedged using the Black-Scholes or Black model. These models assume a one-to-one relation between the option price and the volatility parameter σ. Thus, options prices are often quoted using the implied volatility of the option - the unique value of the volatility which yields the option price when used in these models. The Black-Scholes option pricing model assumes that the volatility of the underlying is constant. In particular, the Black-Scholes model assumes that the volatility of the underlying is unaffected by changes in the price of the underlying over the life of an option or derivative. However, this is contrary to what we see in the market - as the underlying price falls, volatility will tend to rise and vice versa. In reality, options with different strikes require different volatilities σ to yield their correct market prices. This is what is known as the market skew or smile (Dupire, 1994).

Stochastic volatility models are one approach to resolve this shortcoming (the assumption that the volatility of the underlying is constant) of the Black-Scholes model. Rather than assume constant volatility, stochastic volatility models assume that the volatility of the underlying follows a random process. In allowing for this stochastic volatility process, more accurate modelling of derivatives can be achieved. This process is governed by a set of parameters which capture or affect certain aspects of the skew - for example, curvature, slope or level.

One then comes to the issue of calibration - finding the values of these model parameters. The parameters are found by calibrating to available market data. The performance of the calibrator can be evaluated on different criteria. The three criteria with which this thesis is concerned are: the stability over time of the parameters produced, the computational efficiency of the calibrator and the quality of fit of the results to the market data.

If parameters are stable over time, big changes in market data may result in big changes in parameter values but small changes in market data should not result in big changes in parameter values. The stability of model parameters is desirable for a number of reasons. Firstly, in the finance world, models are used to determine asset pricings, holdings and portfolios (be they for speculation or hedging). If parameters vary drastically from day to day, the asset prices and portfolios will vary drastically from day to day, resulting in trading in and out of positions which can lead to high trading costs. Secondly, if the parameters are known to be more stable, they will need to be recalibrated on a less frequent basis. In less liquid markets, where fewer trades are available, it is preferable to rebalance portfolios less frequently as trading may be expensive. Conversely, there may also not be enough new information available to recalibrate frequently.

Computation time can vary depending on the type of calibration scheme and optimiser used.

It is obviously desirable to have a reasonable fit to the market data. However, one needs to be careful to avoid overfitting. Overfitting occurs when the model is so closely fitted to a set of data points that it describes the random error or noise in the data rather than the underlying relationship between the variables. This can reduce the predictive power of the model. Overfitting can be the result of over-parameterisation of the model. Thus, in finding the model parameters, the aim should not be to blindly find a set of parameters which provides the best fit to market data. Rather, a balance should be sought between retaining the underlying relationships between the variables and the fit. The exact nature of this balance will depend on the needs of the practitioner.

This thesis is concerned with the calibration of the SABR and Heston models. Calibration algorithms and methods applied to the SABR model by Hagan (2002) and West (2005) are surveyed. A calibration of the SABR model, applying these methods, is implemented. The purposes of this implementation are to establish that the parameter results are reasonable (using West's (2005) results) and to briefly examine the fit and the stability of the parameters when these methods are implemented. The primary focus of this thesis is the application of these methods to the Heston model and the effect of these methods on the performance of the Heston calibrator.

Hagan (2002) describes the effects of each of the four SABR model parameters on the skew. He finds that two of the parameters perform a similar function. He finds that by fixing one of these parameters (the β parameter) to a market-consistent, economically justifiable level, the remaining calibrated parameters are more stable over time. West (2005) implements the SABR model in the South African market. He, too, finds that by fixing β, the remaining parameters are more stable. This thesis implements the calibration of the SABR model for fixed β and briefly surveys the fit and stability of the remaining parameters when β is fixed.

We then investigate the fixing of parameters in the calibration of the Heston model. We first examine the effect of each of the Heston parameters on the skew in order to identify any redundant parameters. The economic meaning of these parameters is taken into account and these parameters are fixed at market-consistent levels. We then implement calibrators for the cases where no parameters are fixed and where some parameters are fixed. We then compare the results of the Heston calibrators with fixed parameters and those with no fixed parameters in terms of the three criteria mentioned.

This thesis is organised as follows: we begin by reviewing the context in which the calibrations will take place, the South African market, in §2. In particular, we discuss the data and the option pricing. Next, the calibration problem is defined in §3. In §4, the SABR model, its option pricing formulae and the effects of the SABR parameters on the skew are shown. This is repeated for the Heston model in §5. We then describe the details of the implementation of the calibrators in §6. In §7, we survey the results, fit and stability of the SABR parameters when β is fixed. In §8 the results of the Heston calibrations are discussed. Further work is suggested in §9 and the final conclusion is given in §10. All implementations are coded in Matlab and the code is given in the Appendix.

2 The South African Market

2.1 Option Pricing

West (2005) explains that the options on futures in the South African market are American and fully margined. Whilst the option is American, West (2005) states that it can be shown that it is sub-optimal to exercise either calls or puts early. Therefore, the option pricing formula is similar to the Black formula and the futures price in the Black formula can be replaced with the forward price. The forward price, F, is as such: $F = S\mathrm{e}^{(r-q)\tau}$ where S is the underlying spot price, r is the risk-free interest rate and q is the dividend yield. The appropriate option pricing formula is thus:

$$V_c = FN(d_1) - XN(d_2) \tag{1}$$

$$V_p = XN(-d_2) - FN(-d_1) \tag{2}$$

$$d_{1,2} = \frac{\ln \frac{F}{X} \pm \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}} \tag{3}$$

$$\tau = T - t \tag{4}$$

where V_c/V_p is the price of the call/put, X is the strike price, T is the option maturity, t is the current time, τ the term of the option and $N(x)$ is the cumulative normal distribution function.

2.2 Data

The South African options market is less liquid than more developed markets. Thus, it may be necessary to use trade sets (for example, spreads and butterflies) in addition to single option trades as data when calibrating (as explained by West (2005)). Furthermore, it is necessary to use historical option trade data (as opposed to simply calibrating to the current day's option trades).

Within the South African Market, the most liquid options on futures are the options with the Futures Contract on the FTSE/JSE Top 40 Index as underlying. The Top 40 Index is an index of the 40 largest shares. Expiries for these options occur on the 3rd Thursday of March, June, September and December (or on the previous business day if there is a public holiday). The options are pure futures-style American options - the premium will be paid over the life of the option. (See http://www.jse.co.za/Libraries/Equity_Derivatives_-_specifications/Index_Options_Contract_Specifications.sflb.ashx for more detail on the options).

The data for the options contracts expiring in December 2011 is used in this thesis. Sets of options with this expiry traded more than 80 times over the year before expiry. The data set contains option data from 12 October 2010 to 19 October 2011. The option data was obtained from an asset manager.

An example of the data is given in Table 1. The data supplied includes the strike of the option, the forward price (F), the number of contracts sold (Size), the implied volatility of the option (Vol), the term of the option at the time of the trade (in other words, the time between when the trade occurred and the expiry of the option/set), whether the option was bought or sold (B/S), the at-the-money volatility (AtmVol), whether the option was a put or a call (C/P), the days since the trade date (dft), the identity number of the trade (ID) and the date of the trade (TradeDate).

Strike	F	Size	Vol	Term	B/S	AtmVol	C/P	dft	ID	TradeDate
26000	27391	2500	0.246	1.1753	-1	0.240699	-1	506	160000126	12-Oct-10
26000	27559	500	0.25	1.1315	1	0.235769	-1	490	160000159	28-Oct-10
27000	27782	500	0.272	1.0410	1	0.242273	-1	457	160000206	30-Nov-10
26000	28565	1000	0.265	1.0246	1	0.249234	-1	451	160000214	06-Dec-10
29000	28565	1750	0.237	1.0246	-1	0.249234	-1	451	160000214	06-Dec-10
23000	28565	1000	0.221	1.0246	1	0.249234	1	451	160000214	06-Dec-10
26000	28626	2000	0.263	1.0246	1	0.249234	-1	451	160000214	06-Dec-10

Table 1: Example of data

Options with the same identity number form part of a trade set. In the case of sets of options, a price is quoted for each option making up that trade set. However, it is important to note that it is the price of the set as a whole which is traded, not the prices of the individual options. The prices quoted for the individual options making up a trade set in the table above, are quoted by the trader. Thus, a trader can quote any number of combinations of individual option prices which make up the price of the trade set. For example, if a spread is traded at R3000, there are a number of combinations of prices of individual options which could add up to R3000 (e.g. a call costing R1400 and a second call costing R1600; or two calls costing R1500 each). Traders will often misspecify the price of the individual options making up the trade set in order to create such confusion.

This is relevant as the implied volatility of an option is backed out of the Black-Scholes formula and is equivalent to the price of the option. Therefore, different quoted prices imply different implied volatilities. Using the implied volatility to calibrate the parameters could result in incorrect parameter values and option prices. Therefore, it is very important to note that the traded price of the entire set is the only reliable source of information and the quoted implied volatilities cannot be taken at face value.

From this point in the thesis, trade packages will be used to describe the option data - a package can be either a single option or it can be a trade set. For example, if two single options and a spread are traded on a day, we would say that three packages were traded on that day.

3 Calibration problem

3.1 Finding a parameter set

The most common method of finding a suitable parameter set (and the one employed in this thesis), is to find those parameters which produce the correct market prices of vanilla options. These parameter sets can then be used to price more exotic options.

The most popular approach, as used by West (2005) and Moodley (2005), is to minimise the error between the model vanilla option volatilities and the market vanilla option volatilities. The differences between the vanilla option volatilities in the market and those produced by the model are minimised over the parameter space i.e. we find a parameter set $[\sigma_1^{model}, \sigma_2^{model}, \sigma_3^{model}...]$ which minimises f, the error between the fitted market volatility (σ_i^{market}) and the model volatility (σ_i^{model}):

$$f(\sigma_1^{model}, \sigma_2^{model}, \sigma_3^{model}...) = \sum_{i=1}^{n} w(\frac{\sigma_i^{market} - \sigma_i^{model}}{\sigma_i^{market}})^2 \tag{5}$$

where w is some weighting function.

3.2 Fitting in an illiquid market

Given the necessity of using trade sets in the calibration procedure, a slight change in approach to the fitting procedure is necessary. As the package prices are traded (and not the implied volatilities), West (2005) suggests it makes more sense to fit the parameters to the package prices rather than to the option volatilities. In other words, we minimise f, a function of the error between the market price and the model price:

$$f(P_1^{model}, P_2^{model}, P_3^{model}...) = \sum_{i=1}^{n} w(\frac{P_i^{market} - P_i^{model}}{P_i^{market}})^2 \tag{6}$$

where P_i^{market} is the price of package i in the market and P_i^{model} is the price of package i obtained using the model parameter inputs. In the case of P_i^{model}, we obtain the price of a package by finding the prices, under the model, of the individual options making up the package and then adding these prices to obtain the price of the set.

4 SABR Model

4.1 The model

The SABR ("Stochastic alpha, beta, rho") model is as follows:

$$dF = \alpha F^{\beta} dW_1 \tag{7}$$

$$d\alpha = \nu \alpha dW_2 \tag{8}$$

$$dW_1 dW_2 = \rho dt \tag{9}$$

West (2005) and Hagan (2002) describe the variables in the above equations. In the above set of equations, F is the forward price and follows a stochastic process. The parameter α is a stochastic, "volatility-like" parameter which is not equal to the volatility but which has a functional relationship with the at-the-money volatility as is explained in §4.2. The parameter α can also be thought of as the initial volatility. Constant parameter ν is the volatility of volatility parameter. This parameter accounts for the "volatility clustering" which occurs in the market - this is a phenomenon whereby large changes tend to be followed by large changes and small changes by small changes. Parameterβ is an element of the set [0,1]. The closer β is to 1 (0), the more log-normal (normal) is the stochastic model. Furthermore, β determines the relationship between the futures spot and the at-the-money volatility: $\beta \approx 1$ indicates that the person modelling the volatility surface believes that if the market were to move up or down in an orderly fashion, the at-the-money volatility level would not be significantly affected. A value of $\beta < 1$ indicates that if the market were to move, then the at-the-money volatility would move in the opposite direction. The closer β is to 0, the more likely this is to happen. dW_1 and dW_2 are correlated Brownian motions with $-1 < \rho < 1$ being the correlation coefficient parameter. These four parameters need to be found through calibration.

4.2 Option Pricing formula

An appealing feature of the SABR model is that prices of vanilla options can be recovered from the model in closed form. Given the parameters (α, β, ρ and ν), the market inputs (forward price F, strike X) and the remaining time to maturity τ, the option price can be found with the Black formula, equations (1) to (4), by using the correct implied volatility.

In the case of the SABR model, West (2005) and Hagan (2002) suggest that the correct volatility to input into the Black formula is as follows:

$$\sigma(X,F) = \frac{\alpha(1 + (\frac{(1-\beta)^2}{24}\frac{\alpha^2}{(FX)^{1-\beta}} + \frac{1}{4}\frac{\rho\beta\nu\alpha}{(FX)^{(1-\beta)/2}} + \frac{2-3\rho^2}{24}\nu^2)\tau)}{(FX)^{(1-\beta)/2}[1 + \frac{(1-\beta)^2}{24}\ln^2\frac{F}{X} + \frac{(1-\beta)^4}{1920}\ln^4\frac{F}{X}]}\frac{z}{\chi(z)} \tag{10}$$

$$z = \frac{\nu}{\alpha}(FX)^{(1-\beta)/2}\ln\frac{F}{X} \tag{11}$$

$$\chi(z) = \ln(\frac{\sqrt{1-2\rho z + z^2} + z - \rho}{1-\rho}) \tag{12}$$

The at-the-money volatility, $\sigma_{atm} = \sigma(F,F)$ can be found as:

$$\sigma(F,F) = \frac{\alpha(1 + (\frac{(1-\beta)^2}{24}\frac{\alpha^2}{F^{2-2\beta}} + \frac{1}{4}\frac{\rho\beta\nu\alpha}{F^{1-\beta}} + \frac{2-3\rho^2}{24}\nu^2)\tau)}{F^{1-\beta}} \tag{13}$$

Obtaining the α parameter

The α parameter is calibrated to the at-the-money volatility. The following trinomial equation is obtained by inverting equation (13):

$$\frac{(1-\beta)^2\tau}{24F^{2-2\beta}}\alpha^3 + \frac{\rho\beta\nu\tau}{4F^{1-\beta}}\alpha^2 + (1 + \frac{2-3\rho^2}{24}\nu^2\tau)\alpha - \sigma_{atm}F^{1-\beta} = 0 \tag{14}$$

West (2005) suggests using the Tartaglia method to solve this equation.

Obtaining the β parameter

The β parameter can be found using equation (13). Hagan (2002) argues that the second term in the numerator is negligible and thus β can be obtained using the log-log plot:

$$\ln\sigma(F,F) = \ln\alpha - (1-\beta)\ln F \tag{15}$$

4.3 Effect of parameters on skew

This section investigates the effect of each of the SABR model parameters on the skew. For each parameter, the implied volatility has been calculated given different levels of the strike and different levels of that parameter. The implied volatilities for an option with a maturity of one year are shown but the results hold for all maturities (see §A.2. for the code used to generate these graphs).

Alpha (α)

As can be seen in Figure 1, an increase in α shifts the implied volatility skew upwards. A change in α has little effect on the shape of the skew.

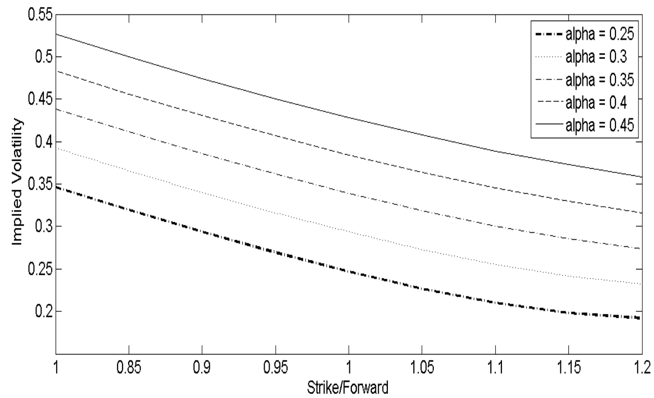

Figure 1: Effect of α (alpha) on skew

Volatility of volatility (ν)

As can be seen in Figure 2, an increase in the volatility of volatility parameter increases the curvature of the volatility skew. A change in ν has little effect on the level of the skew.

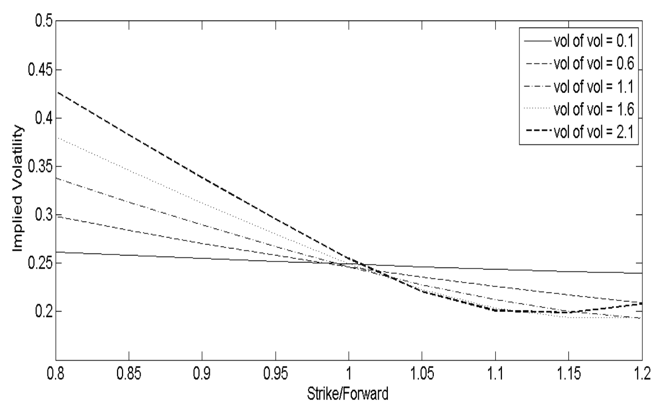

Figure 2: Effect of volatility of volatility, ν, on skew

Correlation Coefficient (ρ)

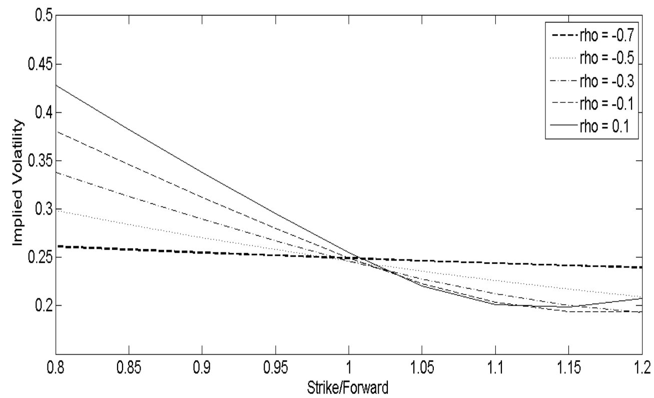

Figure 3: Effect of correlation, ρ (rho), on skew

As can be seen in Figure 3, a change in the correlation coefficient,ρ, changes the slope of the skew.

Beta(β)

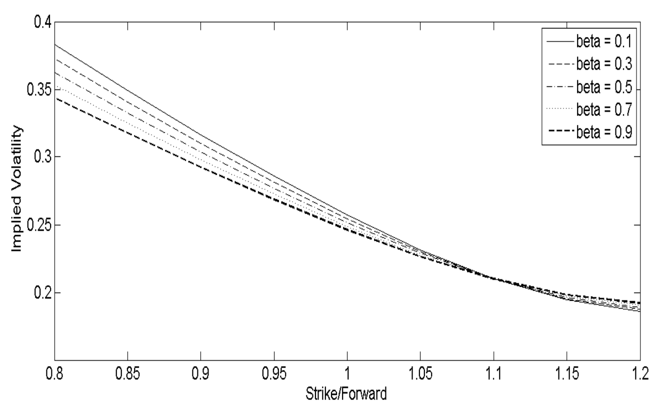

Figure 4: Effect of β (beta) on skew

As can be seen in Figure 4, a change in β affects the slope of the skew.

4.4 Comments on fixing parameters

The maps shown in §4.3 concur with the findings of Gauthier et al (2009), as well as those of Hagan (2002).

Hagan (2002) posits that the value of β has little effect on the goodness of fit. As discussed by both Gauthier et al (2009) and Hagan (2002) and shown in Figures 3 and 4, β has the same effect on the skew as the correlation parameter ρ. Given that the correlation parameter will change the slope of the implied volatility curve, it is prudent to fix β at a certain level. The level of β can be chosen to reflect the practitioner's belief of the normality/log-normality of the market. West (2005) and Hagan (2002) posit that when looking at a time series of the parameters, fixing β to its initial value (the first value of β in the time series) over the period in question, led to the remaining parameters being more stable over the period. Hagan (2002) elaborates that this stability is due to the fact that each of the unfixed parameters have separate roles (as demonstrated in §4.3). In §7, we confirm the stability of the parameters given a fixed β.

5 Heston Model

5.1 The model

The Heston model (developed by Steven Heston (1993)), assumes the futures price, F, and the volatility of the futures price, obey the following stochastic processes (under the risk-neutral measure):

$$dF = rFdt + \sqrt{V_t}FdW_{1,t} \tag{16}$$

$$dV_t = \kappa(\theta - V_t)dt + \xi\sqrt{V_t}dW_{2,t} \tag{17}$$

$$dW_{1,t}dW_{2,t} = \rho dt \tag{18}$$

r is the risk-free rate. Gauthier et al (2009) define the parameters in equations (16), (17) and (18). V is the variance and the parameter V_0 is the initial variance of the underlying futures contract. The parameter $\kappa > 0$ is the mean reversion rate. κ accounts for the degree of volatility clustering (Moodley, 2005). The constant θ parameter is the long-term equity variance of the underlying contract. The ξ parameter is the volatility of the equity volatility. dW_1 and dW_2 are correlated Brownian motions with $-1 < \rho < 1$ being the correlation coefficient parameter.

5.2 Option Pricing

5.2.1 Option Pricing formula

A semi-analytical solution exists for pricing European calls and puts using Fourier inversion techniques (Heston, 1993). The price of a call, as taken from Kahl et al (2005) can be expressed as:

$$C(F,X,V_0,\tau) = [\frac{1}{2}(F - XP(\tau)) + \frac{1}{\pi}\int_0^\infty (Ff_1 - XP(\dot{\tau)}f_2)du] \qquad (19)$$

where F is the forward price, r is the risk-free rate, $P(\tau)$ is the discount factor to the option expiry date, X is the strike, τ is the time to maturity and where

$$f_1 = \mathrm{Re}(\frac{\mathrm{e}^{-iu\ln XP(\tau)}\psi(u-i)}{iuF}) \text{ and } f_2 = \mathrm{Re}\ (\frac{\mathrm{e}^{\ -iu\ln XP(\tau)}\psi(u)}{iu}) \qquad (20)$$

where

$$\psi(u) = \mathrm{e}^{\ C(\tau,u)+D(\tau,u)V_0+iu\ln\ F} \qquad (21)$$

where the coefficients C and D are solutions of a two-dimensional system of ordinary differential equation of Riccati-type:

$$C(\tau,u) = \frac{\kappa\theta}{\omega^2}((\kappa - \rho\omega ui + d(u))\tau - 2\ln\ (\frac{c(u)\mathrm{e}^{d(u)\tau} - 1}{c(u) - 1})), \qquad (22)$$

$$D(\tau,u) = \frac{\kappa - \rho\omega ui + d(u)}{\omega^2}(\frac{\mathrm{e}^{\ d(u)\tau} - 1}{c(u)\mathrm{e}^{d(u)\tau} - 1}) \qquad (23)$$

where

$$c(u) = \frac{\kappa - \rho\omega ui + d(u)}{\kappa - \rho\omega ui - d(u)} \qquad (24)$$

$$d(u) = \sqrt{(\rho\omega ui - \kappa)^2 + iu\omega^2 + \omega^2 u^2}. \qquad (25)$$

The only part needing numerical approximation is the integral in equation (19).

5.2.2 Integration scheme

The integration scheme used in this thesis is that of Kahl and Jackel (2005) in their paper "Not-so-Complex Logarithms in the Heston Model" (§A.1). Kahl et al (2005) argue that the Fourier inversion integrals (as is the integral in (19)) used to calculate option prices are prone to numerical instabilities. They compute the Fourier integral with the aid of adaptive Gauss-Lobatto quadrature. Details on this scheme as well as the Matlab code implementing this scheme and the Heston price can be found in §A.3.

5.3 Effect of parameters on skew

This section investigates the effects of each of the Heston model parameters on the skew. Of particular interest is whether the effect of any of the parameters is similar to that of another parameter. For each parameter, the implied volatility has been calculated given different levels of the strike and different levels of the parameter in question. The implied volatilities for an option with a maturity of one year are shown but the results hold for all maturities. These results concur with the findings of Gauthier et al (2009). (Code used to generate these graphs can be found in §A.4).

Parameters V_0 and θ

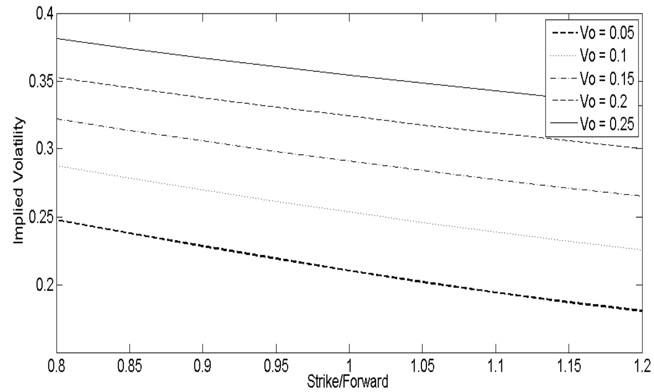

Figure 5: Effect of V_0 on skew

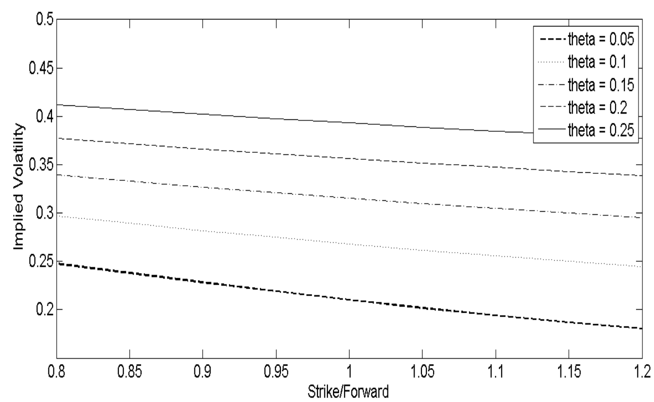

Figure 6: Effect of θ (theta) on skew

As can be seen in Figures 5 and 6, both V_0 and θ affect the level of the skew. An increase in either of these parameters moves the level of the skew upwards.

Parameters κ and ξ

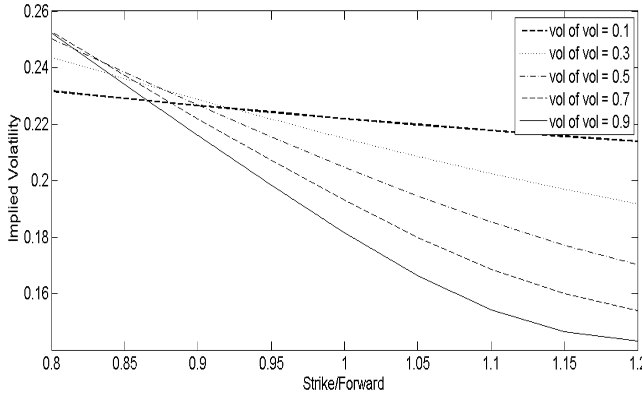

Figure 7: Effect of ξ (vol of vol) on skew

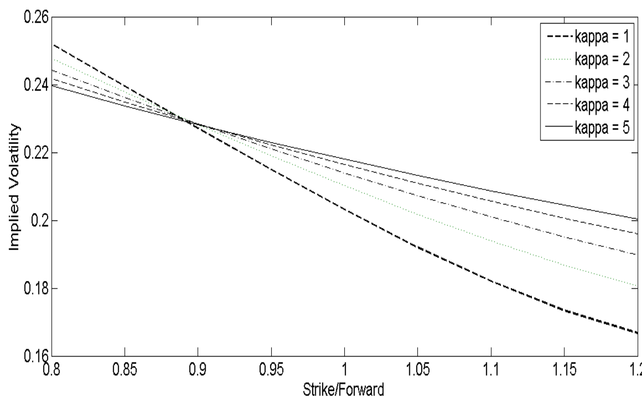

Figure 8: Effect of κ (kappa) on skew

As can be seen in Figures 7 and 8, changes in either ξ or κ affect the curvature of the skew. An increase in κ leads to a flattening of the implied volatility skew, whilst a decrease in ξ has the same effect. Gauthier et al (2009) note that these two parameters have different effects on the level of the skew. However, as V_0 and θ control the level of the skew, this is inconsequential.

Correlation ρ

As can be seen in Figure 9, the correlation parameter controls the slope of the skew.

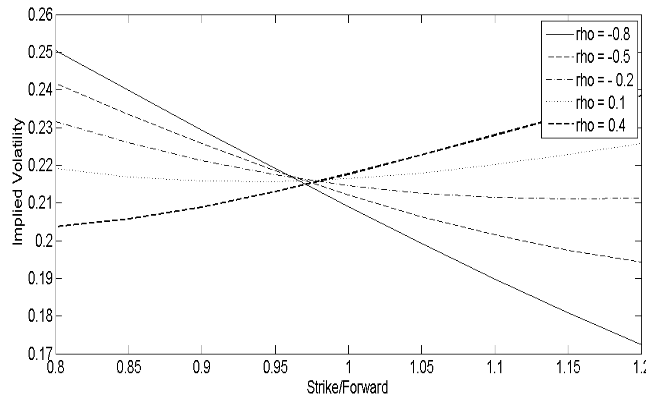

Figure 9: Effect of ρ (rho) on skew

5.4 Comments on fixing parameters

Recall that West (2005) and Hagan (2002) found that two of the SABR model parameters (ρ and β) had the same effect on the skew and that fixing the β parameter led to greater stability of the parameters. A similar approach can be applied to the calibration of the Heston model. Given the similar effects on the skew of two pairs of parameters (the first pair being V_0 and θ, the second pair being κ and ξ), it seems prudent to fix one parameter from each pair (i.e. to fix either V_0 or θ, and to fix either κ or ξ). Thus, rather than calibrating all five parameters, two will be fixed and only three will be calibrated.

Of the first pair of parameters, V_0 or θ, Gauthier and Rivaille (2009) suggest practitioners fix either. V_0 and θ should be set at a level which accounts for the at-the-money variance. Of the second pair of parameters, κ and ξ, they suggest practitioners fix the mean reversion parameter κ. κ should be set to a typical market level.

6 Implementation of Calibration Algorithms

This section describes the implementation of the calibration algorithms for the Heston model. The code for the SABR calibration can be found in §A.5 and for the Heston calibration in §A.6. The code for the time series of errors can be found in §A.7.

6.1 Optimiser `lsqnonlin`

Recall from §3.2 that when one has more traded options/packages than parameters which need to be solved, it is necessary to use an optimisation method to minimise the error between the model price of a package and the market price of a package.

The optimiser used in this thesis is Matlab's `lsqnonlin(fun, x0, lb, ub)`. The inputs to this function are: `fun` is the function to be minimised, `x0` is a vector of the initial parameters, `lb` is a vector of the lower bounds of the parameters and `ub` is the vector of the upper bounds of the parameters. `lsqnonlin` uses an *interior-reflective Newton Method*. To use this method, the nonlinear system of equations cannot be underdetermined. A tolerance of 1e-08 is used.

This is a local optimiser method, the implication of which is that different values of `x0` may lead to different parameter solutions. However, local optimisers are far faster than global optimisers and, in general, perform especially well if the correct initial parameters are chosen (Mikhailov et al, 1993). `lsqnonlin` in particular, has been shown to yield good results (Moodley, 2005). As such, it is popular with practitioners.

6.2 Calibration procedure

In order to compare the performance of the calibrators where no parameters have been fixed and those where parameters have been fixed, we need to observe the parameters produced by the calibrators over an extended period of time. As such, we calibrate frequently over a period of 12 months (the period over which the data spans) in order to produce time series of the parameters. In the case in which parameters are fixed, they are fixed on the first day of the 12 month period and then not changed over that period.

As the data is historical trade data, the value of the parameters will only change when a new trade comes through. If, for example, one calibrates on 1 January 2011 and a new trade was only recorded on the 15 January 2011, the parameters between these times will be constant. Therefore, we have chosen to recalibrate the parameters on every day on which (a) trade(s) occur(s) (in other words, on every new trade-day).

Recall from §2.2 that the data used is for contracts expiring on 15 December 2011. The data contains trade package information from 12 October 2010 to 2 December 2011. This is the period over which we recalibrate. In total, the time series of calibrations consists of 69 calibrations.

The data is a bit sparse for the end of 2010/the beginning of 2011. As such, the first day of our calibration occurs on 10 March 2011. The first calibration will therefore have 14 days worth of trades.

The steps of the calibration procedure for obtaining the parameter set in the case of the calibrator with fixed parameters can be summarised as follows:

- Step 1: Fix chosen parameters on day 1 of time period;
- Step 2: Find the fitted values using `lsqnonlin` on each trade-day for remaining "unfixed" parameters.

In the case of the calibrators with no fixed parameters, step 1 will fall away and fitted values will be found for all parameters on each day using `lsqnonlin`.

6.3 Fixing parameters

6.3.1 SABR

We fix β at 0.7, the value suggested by West (2005).

6.3.2 Heston

Intuitively, it makes sense to fix θ as it is theoretically the long-term variance and as such, we would not expect it to change much over a short time period. Similarly, κ is the mean-reversion speed and we would, in an economic sense, also expect it to remain fairly constant. Thus, in this thesis, θ and κ will both be fixed at appropriate levels. Gauthier et al (2009) suggest fixing κ at 2, a generally accepted market level. We ran the calibrator over the period for both $\kappa = 1$, $\kappa = 2$ and $\kappa = 5$; and the fit was not superior for any of these values. We fix θ at 0.05 which translates to the at-the-money volatility level of about 22% which is reasonable, given our data. More work could be done on at what level to peg these parameters.

6.4 Parameter bounds, initial parameter values and fixed parameter values

`lsqnonlin` requires as inputs `x0`, `lb` and `ub`. The initial values `x0` can affect the outcome of the optimisation. The bounds chosen can also affect the time taken (unnecessarily large bounds can greatly increase computation time). Thus, sensible values for `x0`, `lb` and `ub` must be decided.

SABR Conventional wisdom suggests that the correlation between the change in stock and the change in volatility is negative (Moodley, 2005). In the South African market, the correlation between a rise in stock and a fall in volatility (and vice versa) tends to be rather high. Thus, we set the starting value of ρ to -0.75, the upper bound to -0.02 and the lower bound to -0.99, taking these market levels into account.

The volatility of volatility parameter ν is initially set to 0.8 with an upper bound of 3.5 (it is unlikely to exceed 350%). In free calibrations, the parameter did not go below 0.2 and thus the lower bound is set at 0.2.

Heston In the case where κ is not fixed, the initial value is set to 2, the upper bound to 30 and the lower bound to 0. The initial value of θ is set to 0.05 (in line with the at-the-money volatility level), the upper bound to 0.5 and the lower bound to 0.0025.

We set the initial variance V_0 to 0.05. An upper bound of 0.5 is applied and a lower bound of 0.0025 is applied to ensure V_0 is positive.

In line with our SABR parameter bounds, the volatility of volatility parameter ξ has lower bound 0.2 (in free calibrations, ξ did not go below 0.2), upper bound 3.5 and initial value 0.8. ρ has lower bound -0.99, upper bound -0.02 and initial input -0.75.

6.5 Weighting

Recall from equations (5) and (6), that a weighting w is needed for the function being minimised. An exponentially weighted moving average (EWMA) is chosen to describe the weighting function. The form of the function is λ^y. We chose $y = dft$ where dft is the number of days elapsed since the trade occurred. By using dft in the weighting, trades which occurred more recently are given a greater weighting than older trades. Given the relative illiquidity of the data, a λ of 0.99 was chosen so that the rate of decay of the importance of trades is more gradual. Under this weighting scheme, a trade occuring 3/2/1 months before the calibration date will have a 40%/55%/73% weighting. We ran the calibrations over the period for $\lambda = 0.99$, $\lambda = 0.98$, $\lambda = 0.96$, $\lambda = 0.94$ and $\lambda = 0.9$. The fits for $\lambda = 0.99$ and $\lambda = 0.98$ were much of a muchness, but both were superior to the fits when using a lower λ weighting. This thesis is primarily concerned with the comparison between the performance of the calibrators with fixed parameters and those with no fixed parameters. Thus, consistency between the two calibrators (in terms of the weighting) is the main concern and we found that the choice of λ did not change the conclusions of this thesis.

6.6 Calculating the errors

Recall from §3.2 that the function being minimised is

$$f(P_1^{market}, P_2^{market}, P_3^{market}...) = \sum_{i=1}^{n} 0.99^{dft_i} (\frac{P_i^{market} - P_i^{model}}{P_i^{market}})^2 \tag{26}$$

The error for a single package on any day is

$$0.99^{dft_i} (\frac{P_i^{market} - P_i^{model}}{P_i^{market}})^2 \tag{27}$$

Due to the calibration procedure, different numbers of packages will be calibrated to on different days. In order to make a meaningful time series of errors, we find an average of the percentage differences of all packages calibrated to on a day and define the error for that day as this number. For example, if we calibrate to 20 packages on day X, we will sum the weighted percentage difference of each package and then divide by 20. In other words, the error on any day will be

$$\frac{\sum_{i=1}^{n} 0.99^{dft_i} (\frac{P_i^{market} - P_i^{model}}{P_i^{market}})^2}{n} \tag{28}$$

The primary use of the errors will be to compare the performance of the calibrator when fixing parameters and the calibrator when parameters are not fixed. The errors will also be used to sense-check certain results.

7 Stability of parameters for the SABR calibrator with fixed β

In this section, the parameter values output by the SABR calibrator, the fit and the stability of the parameters for fixed β are surveyed briefly.

7.1 Parameter values

Figure 10 shows the time series of the parameter values for the SABR calibration where β has been fixed. The results seem reasonable given those obtained by West (2005). The ρ and ν parameters are correlated. Some practitioners fix the value of ρ and this might be appropriate in this case, given the correlation between the parameters. Figure 11 shows the parameters when β and ρ are fixed. The value for α for both cases is shown in Figure 12 - they are almost identical.

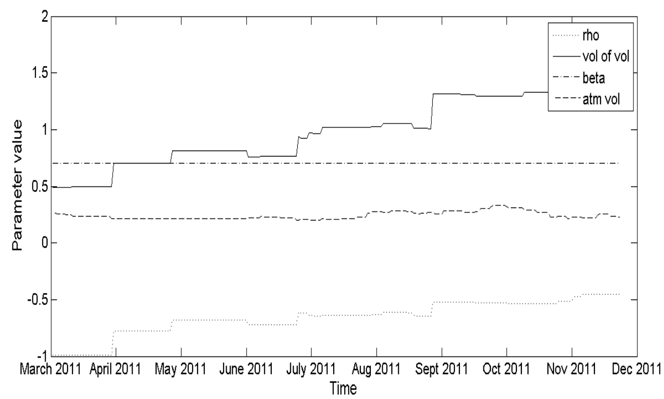

Figure 10: Time series of SABR parameters for β fixed

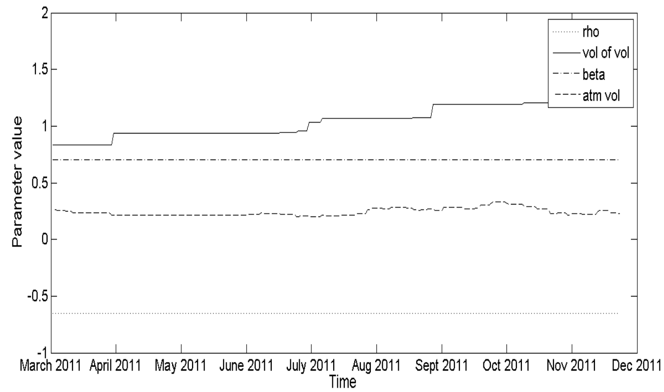

Figure 11: Time series of SABR parameters for β and ρ fixed

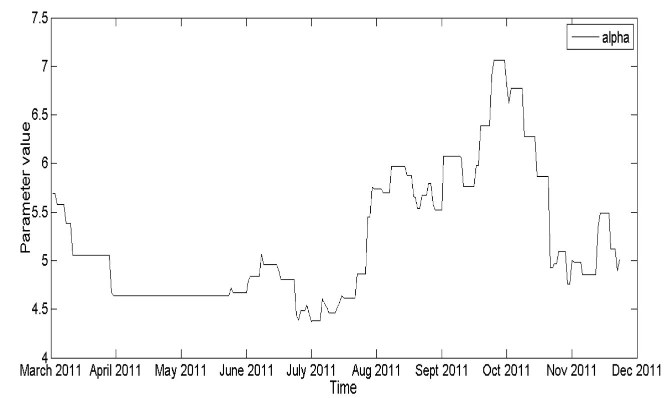

Figure 12: Time series of α parameter

7.2 Errors

Figure 13 shows the time series of errors for β fixed and for β and ρ fixed. The errors increase on 13 April 2011, 12 July 2011, 7 Sept 2011, 19 Oct 2011, 11 Nov 2011. The respective increases in errors correspond with the introduction of trade packages 21, 42, 76, 92, 103 and 107. These packages are all out-the-money packages which appear to be harder to fit.

The errors are, as expected, larger for when only β is fixed than for when β and ρ are fixed.

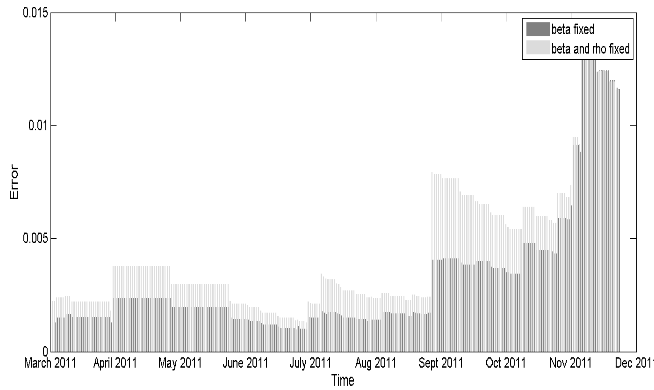

Figure 13: Time series of errors for β fixed versus β and ρ fixed

7.3 Stability of the parameters

Table 2 shows the long-run average of each parameter over the period for β fixed and for β and ρ fixed.

Calibrator	ν	α	ρ
β parameter fixed	1.09	5.25	-0.62
β and ρ parameters fixed	1.08	5.26	-0.65

Table 2: Long-run average of SABR parameters

The changes in α and the at-the-money volatility are shown in Figure **14.** As expected, given the derivation of α (see §4.2), the jumps in α are directly linked to those of the at-the-money volatility.

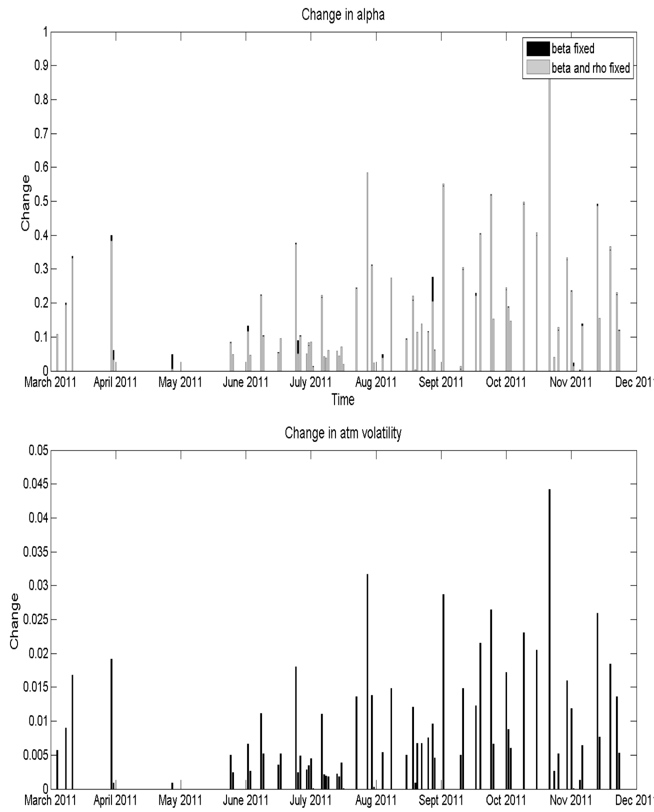

Figure 14: Time series of changes in α and at-the-money volatility (atm vol)

The changes in the ν and ρ parameters are shown in Figure **15**. The largest changes of ν and ρ coincide with the introduction of the packages mentioned in §7.2 (those which caused an increase in error), as well as package 37, which is very out-the-money. The parameters of the Heston model also jump when these trade packages are introduced. The effect of the packages is discussed in more detail in §8.2. With the exception of these jumps, the SABR parameters do not appear to move very much.

From Figure 15, it is apparent that the changes in ν are smaller for β and ρ fixed than for only β fixed.

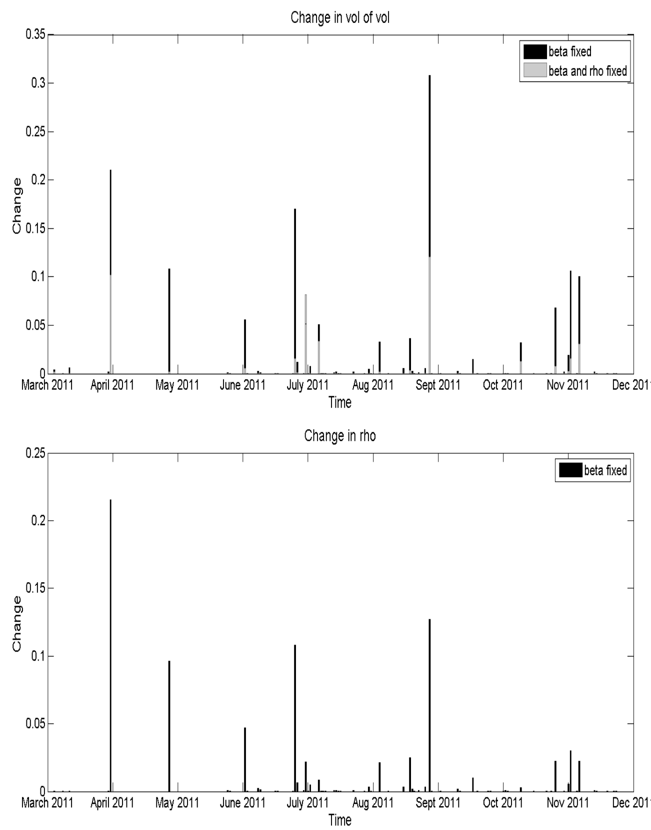

Figure 15: Time series of changes in ν and ρ

Figures 16 and 17 show the rolling standard deviation of ν and α for the calibrator with β fixed and the calibrator with β and ρ fixed. In order to calculate the rolling standard deviation, we calculate the standard deviation over the 20 calibrations up to and including a day. We then move one calibration on in time. Thus, Figures 16 and 17 show the standard deviations on each day a calibration occurs. The first day in this series is 13 July 2011 and there are 48 days in total for which the standard deviation is calculated. The standard deviations of ν over the period are smallest when both β and ρ are fixed. The standard deviations of α are, as expected, much the same for the calibrator with fixed β and that with fixed β and ρ.

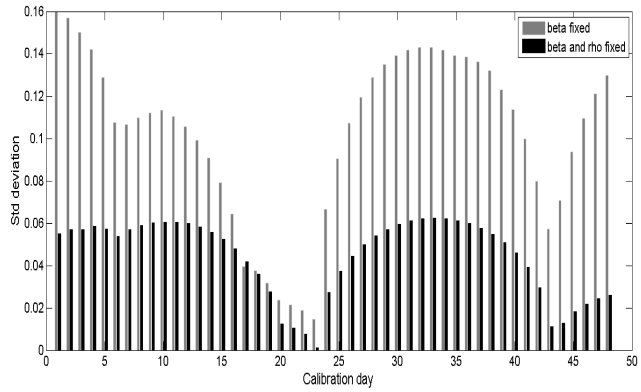

Figure 16: Rolling standard deviation for ν

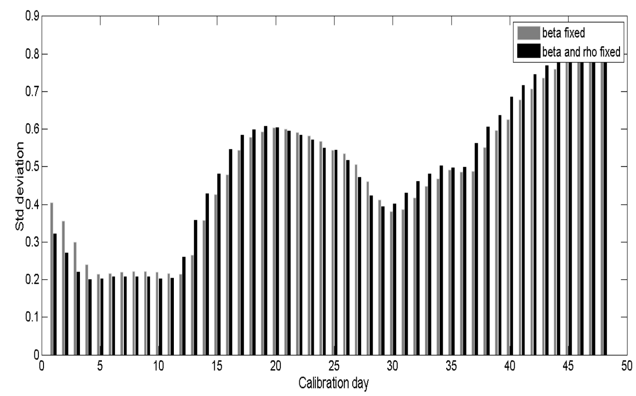

Figure 17: Rolling standard deviation for α

8 Results for the Heston calibrators

In this section, the parameter values output by the Heston calibrators are shown and discussed. The performances of the calibrators with some fixed parameters and of the calibrator with no fixed parameters are compared, based on three criteria. Firstly, the errors of the calibrations are discussed in order to ascertain the quality of the calibrations. Secondly, the stability of the parameters over time are discussed. Lastly, the computation time of the calibrations is compared.

8.1 Parameter values

Figure 18 shows the time series of the calibration parameters where no parameters have been fixed. Figure 19 shows the same time series but with κ removed to show finer detail of the other parameters.

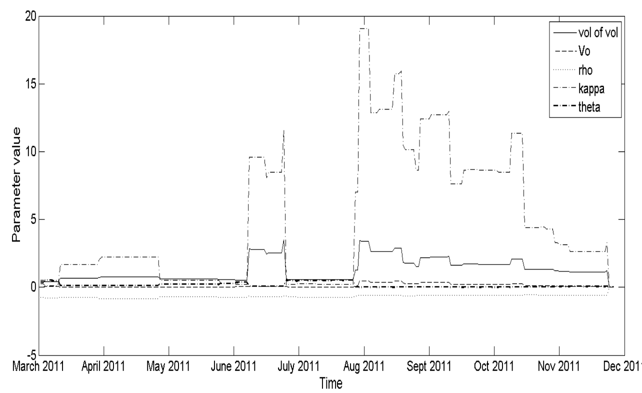

Figure 18: Times series of ξ, Vo, ρ, κ and θ parameters where no parameters have been fixed

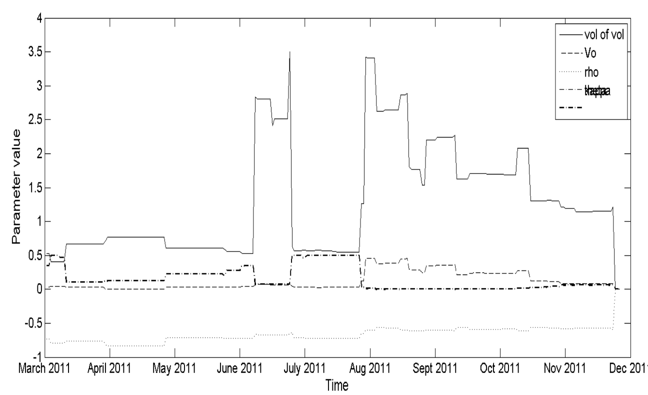

Figure 19: Time series of ξ, Vo, ρ and θ parameters where no parameters have been fixed

Not unexpectedly, the movements of ξ and κ are highly correlated. The high degree of correlation indicates that fixing κ would be justified as both parameters appear to react to the same changes in the market.

From Figure 19 we also observe that V_0 moves with κ and ξ. V_0 is negatively correlated to θ but, surprisingly, not to the same degree as when it is correlated with κ and θ. Recall that V_0 and θ were found to have the same effect on the skew and we chose to fix θ. Incidentally, if only κ is fixed, the correlation between V_0 and θ increases, as does the correlation between ξ and ρ.

Figure 20 shows the parameter values for fixed θ and κ.

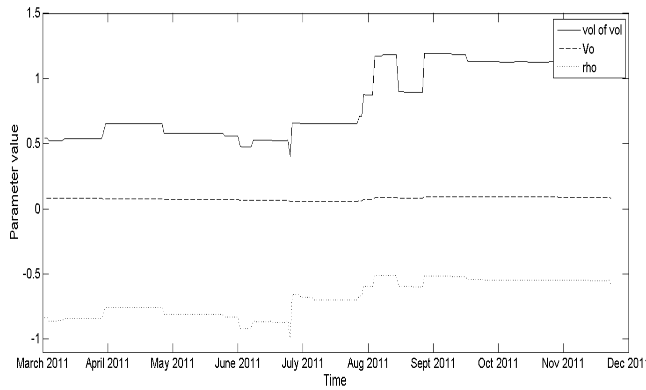

Figure 20: Time series of parameters for θ and κ fixed

Observing Figure 20, we note that the ξ and ρ parameters seem to be correlated once the κ and θ parameters are fixed. This indicates that it may be prudent to fix ρ to an appropriate level. A level of -0.65 is chosen as this is the long-run average of ρ over the time period. Figure 21 displays the time series of the parameters once ρ has been fixed.

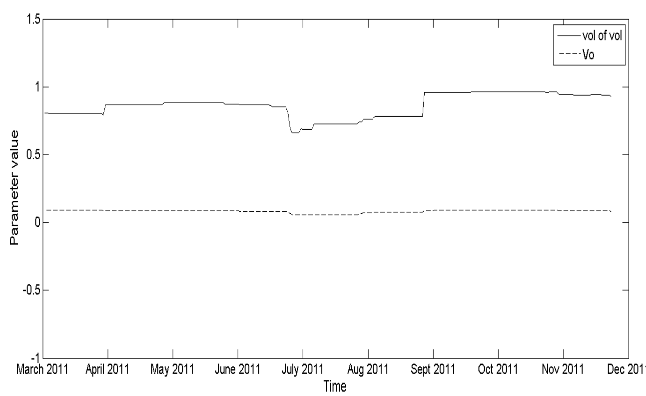

Figure 21: Time series of parameters for θ, κ and ρ fixed

Figure 19 shows, for periods, the θ parameter hits the upper bound of 0.5 and on other days it hits the lower bound of 0.0025 (for the case where θ is not fixed). This is investigated to ascertain the effect on the performance of the calibrator. The relationship between the time series of errors and the time series of θ is examined for each day where a new calibration takes place (i.e. 69 calibration days). The time series of errors when the upper bound of θ is set to 0.5 and to 10 were also compared. The results are shown in Figure 22. The error is relatively low and does not increase greatly when θ hits 0.5 (when the upper bound = 0.5). If the θ parameter hitting the bound affects the performance of the calibrator, we would expect the errors to be lower when the upper bound is equal to 10 than when it is set to 0.5. As can be seen in Figure 22, the time series of errors are identical for both values of the upper bounds over the period where θ hits the upper bound. Thus, increasing the bound above 0.5 seems to have no effect.

From Figure 22 it is apparent that the error increases when θ approaches and eventually hits the lower bound. This increase coincides with the introduction of package 92 - one of the out-the-money packages which caused an increase in error in the SABR calibration. In Figure 18, note that κ increases dramatically as θ decreases towards the lower bound. This is expected - observing equation 17, the κ parameter needs to be high in order to compensate for the low θ.

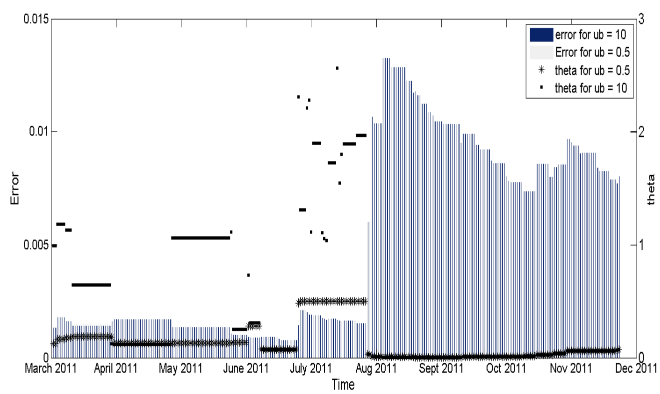

Figure 22: Time series of errors and θ, for θ upper bound = 0.5 versus θ upper bound = 10

In Figure 20 we observe that ρ hits the lower bound of -0.99 on 7 July 2011. Figure 23 shows the time series of ρ versus the errors produced when κ and θ are fixed. The error increases quite significantly. This coincides with the introduction of package 37 - this package was mentioned in §7.3 and is discussed further in §8.2.

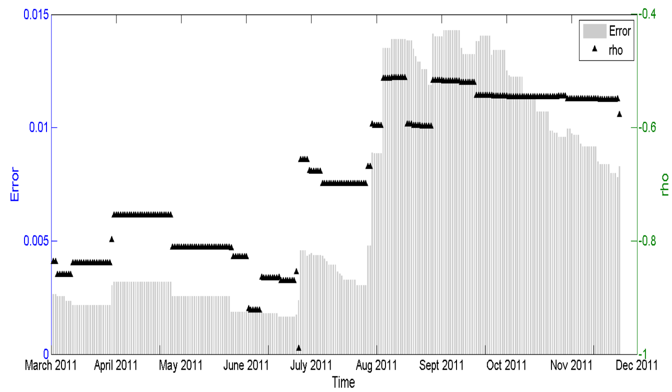

Figure 23: Time series of ρ versus errors for fixed κ and θ

8.2 Errors

Figure 24 shows the time series of errors for the case where no parameters have been fixed, where κ and θ have been fixed and where κ, θ and ρ have been fixed. As this figure shows, the errors follow more or less the same trend with, as expected, the general error level being highest when κ, θ and ρ have been fixed and lowest when no parameters have been fixed. The maximum errors for when κ and θ are fixed (on 19 Aug 2011) and for when no parameters are fixed (on 15 Aug 2011) are very similar. A trader (if being conservative) would probably quote the highest error of the time series as the potential error and thus, in practice, the two time series of errors are not that different. However, the highest error for κ, θ and ρ fixed occurs on 7 Sept 2011 and is much larger than the highest error of the other two time series. Therefore, in practice, the time series of errors for κ, θ and ρ fixed is quite different to those of κ and θ fixed and of no parameters fixed.

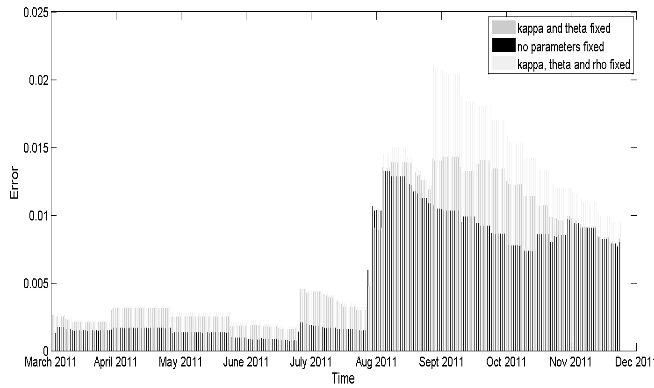

Figure 24: Comparison of time series of errors for the calibrators

The errors seem to be at a fairly low level initially, apart from a marginal increase on 13 April 2011. They increase slightly from 7 July 2011 and increase significantly from mid-August. In the case of κ, θ and ρ fixed, the errors spike on 5 September 2011. They increase marginally at this time for κ and θ fixed. The error drops down for all three cases but increases again around 19 Oct 2011 for the case in which no parameters have been fixed.

These increases coincide with the introduction of packages 21 (13 April 2011), 37 (7 July 2011), 76 (5 Sept 2011) and 92 (19 Oct 2011), respectively. Recall from §7.2, that these packages also caused changes in the SABR parameters and an increase in the error. The increase at mid-August coincides with packages 61 and 64, and a large change in the at-the-money volatility level. This change in at-the-money volatility caused a change in the α parameter in SABR, but not a large increase in the error - the α parameter in SABR is calibrated to the at-the-money volatility. As was mentioned in §7.2, these are out-the-money trades, which would also serve to explain why the error increases when parameters are fixed (they become harder to price with less freedom).

We investigated the errors of these packages in more detail (note only packages 76 and 37 are shown). The time series of the errors of package 76 are shown for each calibrator in Figure 25. In this figure, the error contributed by package 76 to the total error is shown for each calibration over the period. The contribution of error by package 76 is, relatively, quite large for κ, θ and ρ fixed.

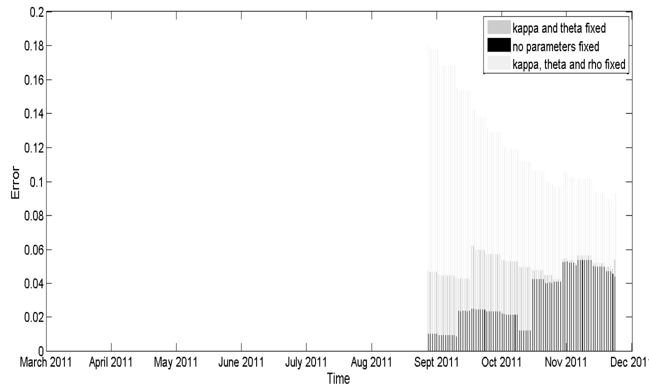

Figure 25: Time series of error contributed by package 76

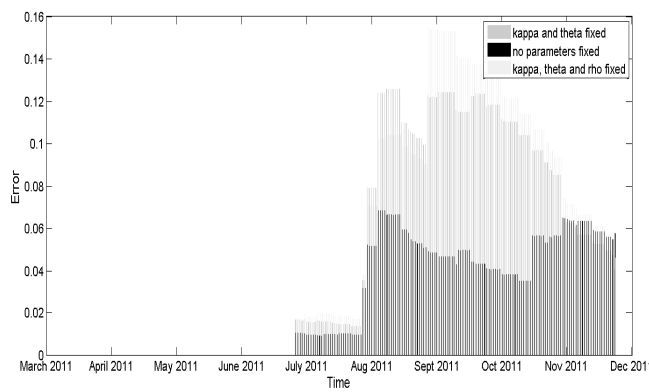

Figure 26: Time series of error contributed by package 37

The time series of the errors of package 37 are shown for each calibrator in Figure 26. The time series of the error for package 37 is very similar to the time series of the total errors (Figure 24).

Figure 27 shows the errors contributed by each package calibrated to on the last day in the time series (2 December 2011) for the calibrators where no parameters have been fixed, where κ and θ have been fixed and where κ, θ and ρ have been fixed. Package 37 seems to contribute quite largely to the error despite being traded in July and having a weighting of 23% (as $\lambda = 0.99$). Whilst only the package errors for 2 Dec 2011 are shown, package 37 had relatively large errors on most days as can be seen in Figure 26.

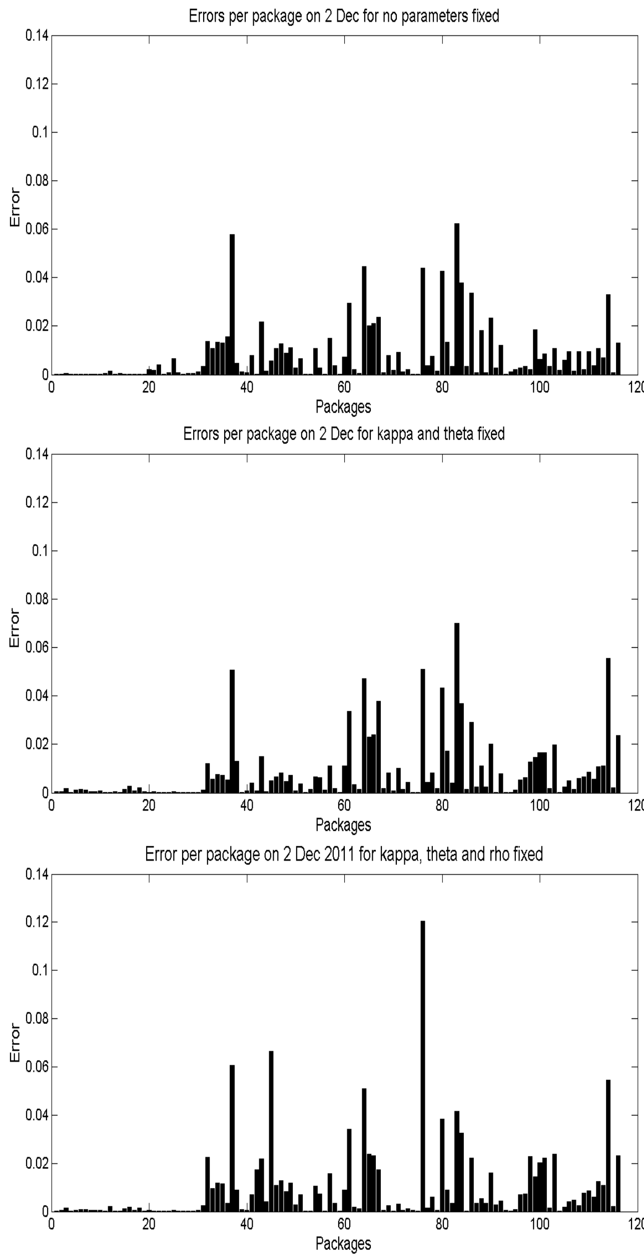

Figure 27: Errors contributed by each package on 2 Dec 2011 for calibrators with fixed and no fixed parameters

This could perhaps be remedied by adjusting the weighting in the calibrator - a lower value of λ could help to reduce the impact of earlier trades. One would need to decide to what extent past data should be included in the calibration procedure versus the errors which are incurred. Weighting by volume could also help with this problem - this is a lower volume, single option trade. Another alternative is to eliminate this package from the data to which one is calibrating.

8.3 Stability of parameters over time

The diagrams in Figure 28 show the time series of the ξ, V_0 and ρ Heston parameters for the case where no parameters have been fixed, where κ and θ have been fixed and where κ, θ and ρ have been fixed.

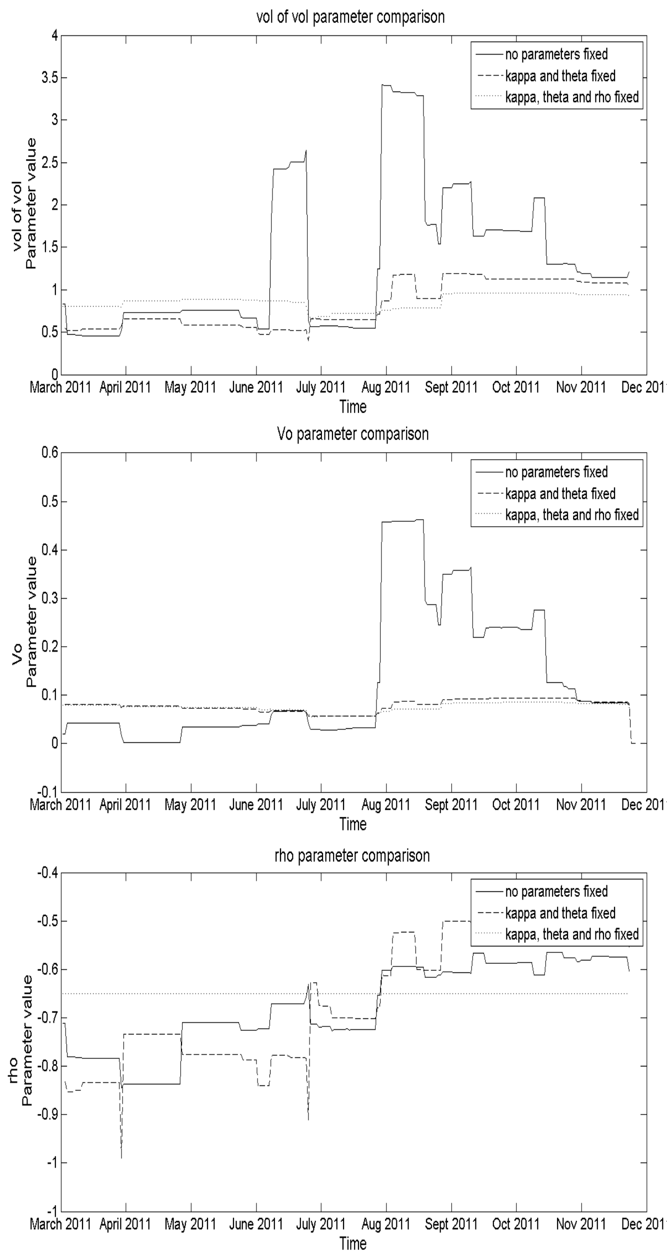

Figure 28: Time series of parameters for no parameters fixed versus θ and κ fixed versus θ, κ and ρ fixed

As can be seen from these figures, the ξ and V_0 parameters appear to be more stable over time if the κ and θ parameters are fixed. The ρ parameter, however, does not appear to be more stable.

Table 3 shows the long-run average of each parameter for the calibrators with fixed and no fixed parameters.

Calibrator	ξ	V_0	ρ	κ	θ
No parameters fixed	1.4	0.14	-0.65	5.48	0.17
θ and κ parameters fixed	0.85	0.08	-0.66	2.0	0.05
θ, κ and ρ parameters fixed	0.85	0.08	-0.65	2.0	0.05

Table 3: Long-run average of Heston parameters

Figures 29, 30 and 31 show the changes in value from one calibration to the next for each of the parameters. In the case of ξ and V_0, the changes when no parameters are fixed are so much larger than the changes when parameters are fixed that they dominate the figure. Thus, we have included figures in which the changes in ξ and the changes in κ are compared only for the cases of κ and θ fixed and κ, θ and ρ fixed (note the differing scales in these figures). In general, the largest differences over the period are smallest when more parameters are fixed and largest when no parameters are fixed. The exception to this is ρ - the change is ρ is substantially larger when κ and θ are fixed than when no parameters are fixed for, in particular, the days on which package 21, 37 and 76 are introduced. In the absence of the freedom afforded by κ and θ, larger changes in ρ are necessary in order to price these very out-the-money trades.

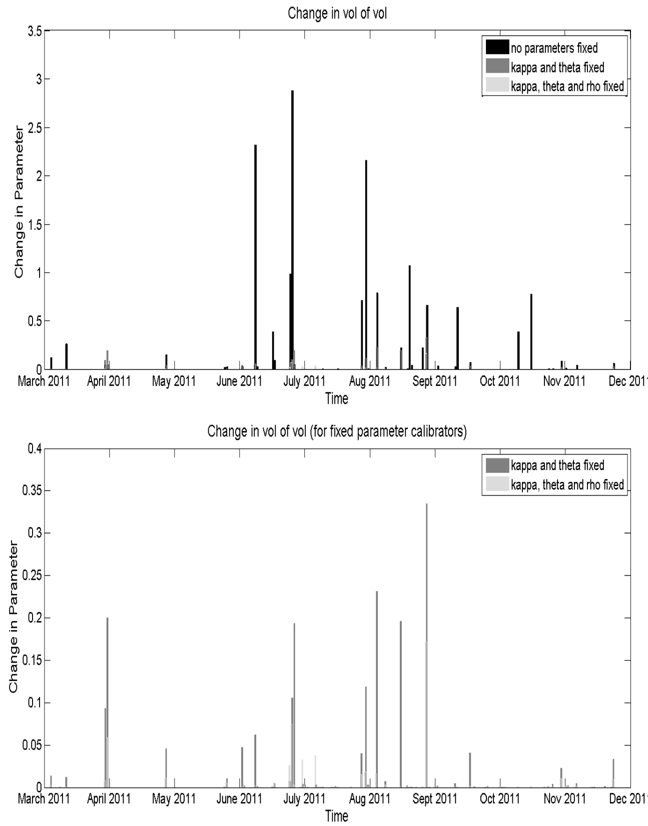

Figure 29: Change in ξ

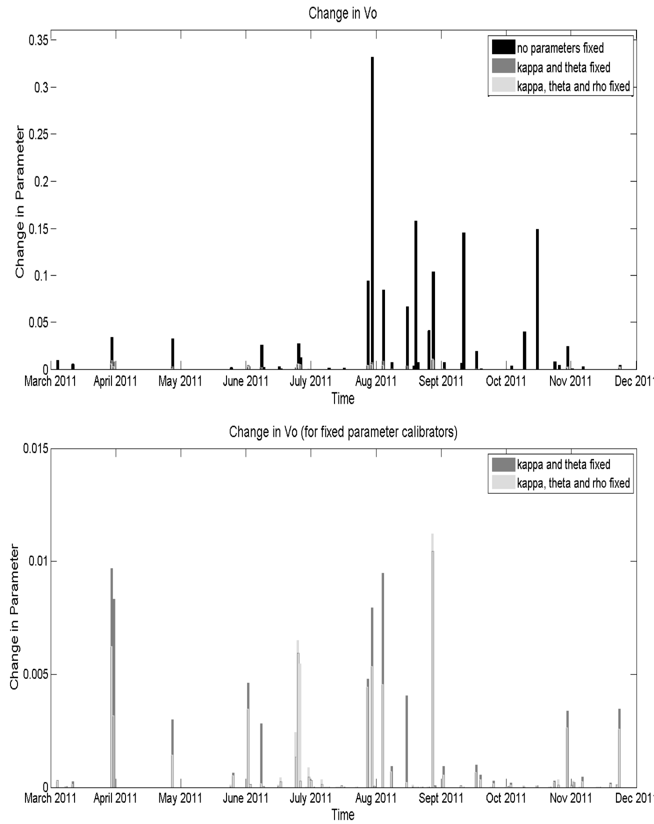

Figure 30: Change in V_0

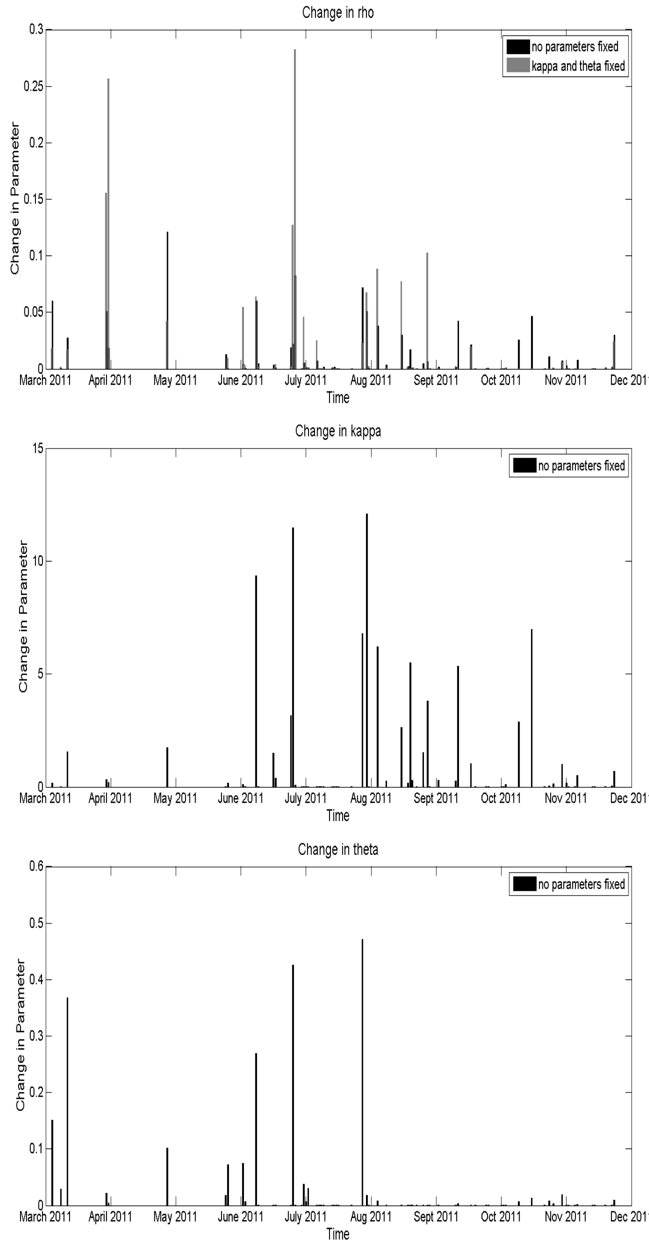

Figure 31: Change in ρ, κ and θ

Figures 32, 33, 34, 35 and 36 show the rolling standard deviation (over 20 calibrations) of each parameter for the calibrators with κ and θ fixed; κ , θ

and ρ fixed and no parameters fixed. Whilst most of the standard deviations over the period are smallest when three parameters are fixed and largest when no parameters are fixed, we note this is not the case for ρ, which confirms the results of the changes of ρ. The standard deviations over the period for V_0 and ξ are significantly greater for no parameters fixed, than for when either κ and θ are fixed or κ, θ and ρ are fixed.

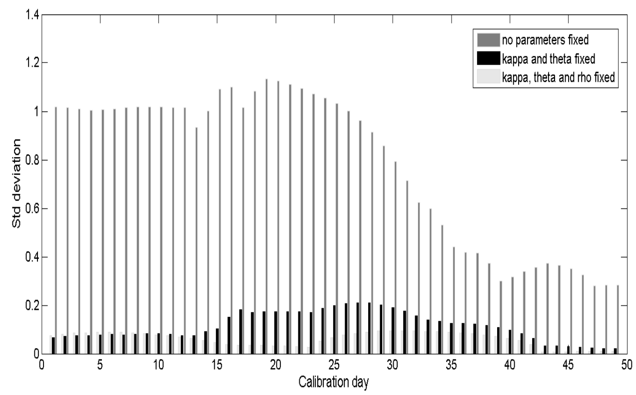

Figure 32: Rolling standard deviation of ξ

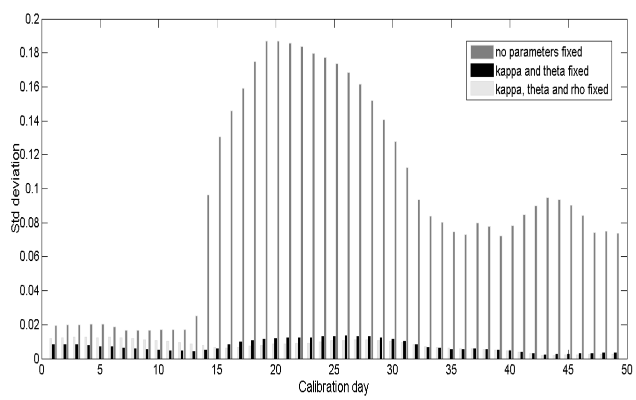

Figure 33: Rolling standard deviation of V_0

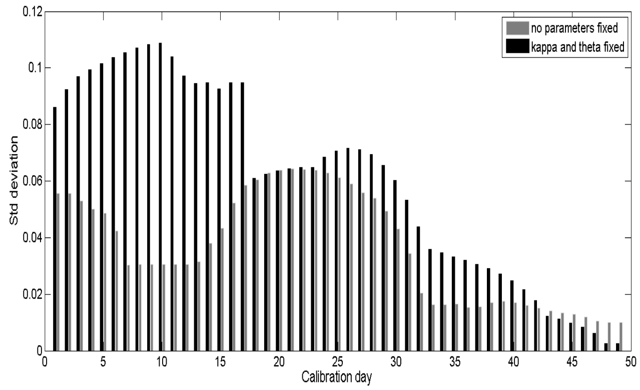

Figure 34: Rolling standard deviation of ρ

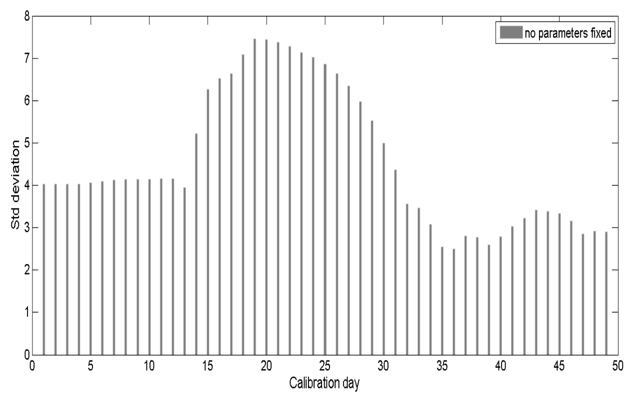

Figure 35: Rolling standard deviation of κ

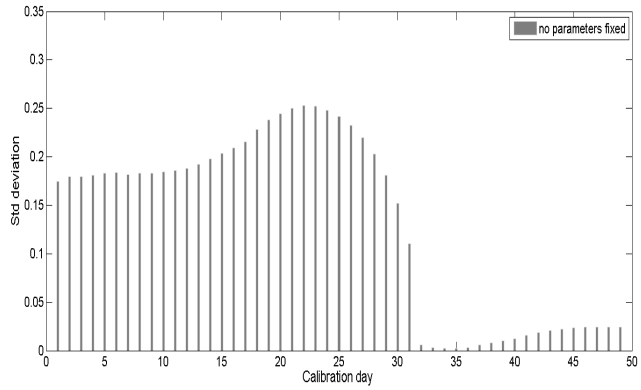

Figure 36: Rolling standard deviation of θ

8.4 Computation time

Table 4 shows the computation time for the calibrator in which no parameters are fixed, the calibrator in which κ and θ are fixed and the calibrator in which 3 parameters are fixed. This calibration takes place on the last day of the data set (2 December 2011).

Calibrator	Computation time (in seconds)
No parameters fixed	458.02
θ and κ parameters fixed	64.4
θ, κ and ρ parameters fixed	27.84

Table 4: Computation time for the different Heston calibrators

Fixing 3 or even 2 of the parameters significantly reduces the computation time.

9 Further work and possible improvements

This thesis has considered limited values for fixed κ, θ and ρ. More extensive work can be done on finding the most appropriate value for each of these parameters.

An intermediate approach can also be explored: rather than fixing a parameter, small bounds can be applied when running the optimiser to calibrate the parameters. The value to which one would fix the parameter (were one fixing the parameter) should fall within these bounds. The expected result would be a better (worse) fit with slightly less (more) stability than the case in which the parameter is fixed (unfixed). The choice of the intermediate approach versus that of fixing parameters would depend on the desired error level.

The fit may be improved by adjusting the weighting, w, in the function to be minimised (equations 5 and 6 with discussion in §6.5). As mentioned in §8.2, different values of λ and their effect on the fit can be investigated. In addition to time, packages could be weighted by traded volume. This would implicitly take into account the liquidity of various trades as options/packages more in-the-money are usually more heavily traded than those out-the-money. In the case in which parameters are fixed, this may have an effect, as the fixing of a parameter decreases the freedom fo fit an illiquid trade. Thus, the more illiquid, out-the-money trades would have a lower weighting and would affect the calibration fit less.

10 Conclusion

When calibrating parameters to market data, it is necessary to find a balance between obtaining a good fit to the market data, the stability of parameters over time and the computation time of the calibration. By investigating the effect of the stochastic model parameters on the skew, one can ascertain which parameters are redundant. Understanding the economic meaning of the parameters allows one to fix the redundant parameters to an appropriate, market-consistent level. Although the fit was slightly worse, we found that fixing redundant parameters in the Heston model led to greater stability over time of most of the remaining parameters and to less computation time.

A Appendix

The appendix contains the code used to generate the data for analysis. All coding is in Matlab.

Please note that comments are prefixed by a '%' sign.

A.1 SABR Option Pricer

The function `Black76` finds the Black (1976) price:

```
function y = Black76(F,K,T,v,PutCall);

%Required inputs:
% F = futures price;
% K = strike price;
% T = maturity in years;
% v = volatility;
% PutCall = 'Call' = 1 or 'Put' = -1;

  d1 = (log(F/K) + (v^2/2)*T)/v/sqrt(T);
  d2 = d1 - v*sqrt(T);
  D1 = PutCall*d1;
  D2 = PutCall*d2;
  y = PutCall*(F* cdf( 'normal',D1, 0, 1) - K* cdf('normal', D2, 0,
1));
end
```

The function `SABRvol` finds the SABR volatility. This can be input into the Black76 function to find the option price under the SABR model. Note that we separate the calculation of the volatility into the at-the-money case (this aligns with equation (13) in the text) and the non-at-the-money case (equations (10), (11) and (12) in the text):

```
% Required inputs:
% a = alpha parameter
% b = beta parameter
% r = rho parameter
% v = vol of vol parameter
% F = spot price
% K = strike price
% T = maturity

function y = SABRvol(a, b, r, v,F,K,T);
  if abs(F-K) <= 0.001                       % Atm vol case.
    Term1 = a/F^(1-b);
    Term2 = ((1-b)^2/24*a^2/F^(2-2*b) + r*b*a*v/4/F^(1-b) + (2-3*r^2)*v^2/24);
```

```
    y = Term1*(1 + Term2*T);
  else                                           % Non-atm vol case.
    FK = F*K;
    z = v/a*(FK)^((1-b)/2)*log(F/K);           % Formula (11)
    x = log((sqrt(1 - 2*r*z + z^2) + z - r)/(1-r));  %Formula (12)
    Term1 = a / FK^((1-b)/2) / (1 + (1-b)^2/24*log(F/K)^2 + ...
(1-b)^4/1920*log(F/K)^4);
    if abs(x-z) < 1e-10
      Term2 = 1;
    else Term2 = z / x;
    end
    Term3 = 1 + ((1-b)^2/24*a^2/FK^(1-b) + r*b*v*a/4/FK^((1-b)/2) +...
(2-3*r^2)/24*v^2)*T;
    y = Term1*Term2*Term3;
  end
end
```

A.2 SABR Sensitivity maps

The following four functions are used to produce the parameter maps in §4.3. The inbuilt Matlab function `blsimpv` backs out the Black Scholes implied volatility from an option price. The parameter being mapped is allowed to vary and the other three parameters are kept fixed. The strike is also allowed to vary.

Note the inputs:

```
% K = strike;
% a = alpha;
% nu = volatility of volatility nu;
% b = beta;
% r = rho;

function res = alphaMap(a, K);
  for i=1:5
    for j= 1:9;
      vol(i,j) = SABRvol(a+0.05*(i-1),0.85,-0.7,1.2,1,(K+ (j-1)*5)/100,1);
      price(i,j) = Black76 (100, K+(j-1)*5, 1, vol(i,j), 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  res = ImVol';
end

function res = nuMap(nu, K);
  for i=1:5;
    for j= 1:7;
```

```
      vol(i,j) = SABRvol(0.25,0.85,-0.7,nu + (i-1)*0.5,1,(K+ (j-1)*5)/100,1);
      price(i,j) = Black76 (100, K+(j-1)*5, 1, vol(i,j), 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  res = ImVol';
end

function res = betaMap(b, K);
  for i=1:5;
    for j= 1:6;
      vol(i,j) = SABRvol(0.25,b+0.2*(i-1),-0.7,1.2,1,(K+ (j-1)*5)/100,1);
      price(i,j) = Black76 (100, K+(j-1)*5, 1, vol(i,j), 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  res = ImVol';
end

function res = RhoMap( r, K);
  for i=1:5
    for j= 1:9;
      vol(i,j) = SABRvol(0.1,0.85,r+ (i-1)*0.2,1.2,1,(K+ (j-1)*5)/100,1);
      price(i,j) = Black76 (100, K+(j-1)*5, 1, vol(i,j), 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  res = ImVol';
end
```

A.3 Heston Option Pricer

As discussed in §5.1, there exists a semi-analytic solution to the Heston option price. All that remains is to solve the integral in equation (19). The scheme chosen to solve this integral is presented in the paper by Kahl et al (2005). Kahl et al (2005) argue that many methods of solving the integral lead to numerical instability which, in turn, results in formulae which are not robust for moderate- to long-dated maturities or strong mean reversion. The problem is that the integrands f_1 and f_2 (equation (20)) are oscillatory in nature and when we evaluate them using a quadrature scheme, discontinuities may arise. This can not only lead to incorrect calculations but also to ones which take a greater amount of time to calculate. Kahl et al (2005) propose their scheme to rectify the instability. The details behind the instability and their solution can be found in detail in their paper.

In terms of evaluating the integral, Kahl et al (2005) use an adaptive Gauss-Lobatto algorithm. The Gauss-Lobatto algorithm is designed to operate on a closed interval $[a, b]$. As such, the authors transform the original integral boundaries $[0, \infty]$ to the finite interval $[0, 1]$. The integral now needs to be evaluated for each $x \epsilon [0, 1]$, rather than for each $x \epsilon [0, \infty]$. Once they have transformed the limits, they evaluate the integral. The integral needs to be evaluated at the new limits $x = 0$ and $x = 1$, as well as over the interval $(0, 1)$. Recall that there can be discontinuities when evaluating the integral. In this case, formulae are given for where the integral is discontinuous over the interval $(0, 1)$.

The following functions implement Kahl et al's (2005) scheme and return the Heston option price. They are best viewed in conjunction with Kahl et al's (2005) paper.

`Heston_Call` returns the Heston call price for the South African market where options are American and fully margined (see §2.1). It utilises the function `HestonQuad` to evaluate the integral in (19).

`HestonQuad` uses the in-built Matlab function `quadl` to approximate the integral from $[0, 1]$ (`quadl(fun,a,b)` approximates the integral of function `fun` from `a` to `b`, to within an error of 10-6 using recursive adaptive Lobatto quadrature). The function being integrated is `con_y`.

Function `con_y` returns the value of a function at point x. It specifies formulae for the limits of $x = 1$ or $x = 0$. To calculate the integral over $x \epsilon (0, 1)$, the function `discon_y` is called.

Functions `f1` and `f2` are called in `discon_y` and function `phi` is called in both `f1` and `f2`.

```
function ret = Heston_Call( F, K, kappa, rho, theta, tau, Xi, Vo, interest,
PutorCall)
  y = HestonQuad(F,K, kappa, rho, theta, tau, Xi,Vo);
  if PutorCall == 1
      ret = y
  else
      ret = y + K - F
  end
end

function ret = HestonQuad(F, K , kappa, rho, theta, tau, Xi, Vo)
  ret = quadl(@(x)con_y (x, F, K, kappa, rho, theta, tau, Xi, Vo),
0, 1);
end

function ret = con_y(x, F, K, kappa, rho, theta, tau, Xi, Vo )
  for i = 1:  length(x)
    xPrime = x(i);
```

```
    if abs(x) <= exp(-50)         %values at limit x = 0
      ret = 0.5* (F - K);
  elseif abs(x-1) <= exp(-50)         % values at limit x = 1
    lnfk = log(F/K);
    IMCf1 = 0;
    IMDf1 = 0;
    if abs(kappa - rho*Xi) <= exp(-50)
      IMCf1 = kappa*theta*tau*tau/4;
      IMDf1 = tau/2;
    else
      IMCf1=(theta*kappa*(exp(-(kappa-rho*Xi)*tau)+...
((kappa-rho*Xi)*tau-1)))/(2*((kappa-rho*Xi)^2));
      IMDf1 = (1- exp(-(kappa - rho*Xi)*tau))/(2*(kappa-rho*Xi));
    end
    limitf1 = lnfk + IMCf1 + IMDf1*Vo;
    IMCf2 = - (exp(-kappa*tau)*theta*kappa + kappa*theta*...
(kappa*tau - 1))/2/kappa/kappa;
    IMDf2 = - (1 - exp(-0.5*kappa*tau))/2/kappa;
    limitf2 = lnfk + IMCf2 + IMDf2*Vo;
    C_Infty = sqrt(1 - rho*rho)/Xi*(Vo + kappa*theta*tau);
    yAtOne = 0.5*(F -K) + (F * limitf1 - K*limitf2)/pi/C_Infty;
    ret = yAtOne;
  else                                    % Finds value over xϵ(0,1).
    yValue = discon_y(xPrime, F, K, kappa, rho, theta, tau, Xi, Vo);
    ret(i) = yValue;
  end
end

function ret= discon_y( x, F, K, kappa, rho, theta, tau, Xi, Vo)
  C_Infty = sqrt(1- rho*rho)/Xi*(Vo+kappa*theta*tau);
  arb = -log(x)/C_Infty;
  F1 = f1(-log(x)/C_Infty, Vo, F, K, Xi, kappa, rho, tau, theta);
  F2 = f2(-log(x)/C_Infty, Vo,F, K, Xi, kappa, rho, tau, theta);
  value = 0.5*(F-K) + (F.*F1-K.*F2)./(x*pi.*C_Infty);
  ret = value';
end

function ret = f1(u, Vo, F, K, Xi, kappa, rho, tau, theta)
  Phi = phi(u - 1i, Vo, F, K, Xi, kappa, rho, tau, theta);
  f = real((exp(-1i.*u * log(K)).* Phi./ (1i.*u*F)));
  ret = f;
end

function ret = f2(u, Vo, F, K, Xi, kappa, rho, tau, theta)
  Phi = phi(u, Vo, F, K, Xi, kappa, rho, tau, theta);
  f = real((exp(-1i.*u .* log(K)).* Phi./ (1i.*u)));
```

```
  ret = f;
end

function ret = phi( u, Vo, F, K, Xi, kappa, rho, tau, theta )
  iu = i * u;
  d = sqrt((rho*Xi.*iu - kappa).*(rho*Xi.*iu - kappa) + ...
Xi*Xi.*(iu + u.*u));
  c = (kappa - rho*Xi*iu + d)./(kappa - rho*Xi*iu - d);
  tc = angle(c); GD = c - 1;
  m = floor((tc + pi)/ (2*pi));
  GN = c.* exp(d.*tau) - 1;
  n = floor((tc+imag(d)*tau + pi)/2/pi);
  lnG = log(abs(GN./GD)) + i*(angle(GN)-angle(GD) + 2*pi*(n - m));
  D = (kappa - rho*Xi.*iu +d)./(Xi*Xi).*(exp(d.*tau) -1)./...
(c.*exp(d.*tau) - 1);
  C = kappa*theta/(Xi*Xi)*((kappa-rho*Xi.*iu + d).*tau - 2.*lnG);
  phi = exp(C+D*Vo+iu.*log(F));
  ret = phi;
end
```

A.4 Heston Sensitivity maps

The following five functions are used to produce the parameter maps in §4.3. The inbuilt Matlab function `blsimpv` backs out the Black Scholes implied volatility from an option price. The parameter being mapped is allowed to vary and the other three parameters are kept fixed. The strike is also allowed to vary.

Note the inputs:

```
% K = strike;
% kappa = mean reversion parameter;
% xi = volatility of volatility;
% theta = long-term volatility;
% rho = correlation parameter;
% InVol = Initial Volatility

function ret = InitialVolMap(InVol, K )
  for i=1:5;
    for j= 1:9;
      price(i,j) = Heston_Call(100, K+(j-1)*5,2, -0.7, 0.05, 1, 0.4,
InVol + 0.1*(i-1), 0, 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  ret = ImVol';
end
```

```
function ret = kappaMapH(kappa, K )
  for i=1:5;
    for j= 1:9;
      price(i,j) = Heston_Call(100, K+(j-1)*5, kappa + 0.5*(i-1), -0.7,
0.05, 1, 0.4, 0.05, 0, 1)
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  ret = ImVol';
end

function ret = RhoMapH( rho, K )
  for i=1:5;
    for j= 1:9;
      price(i,j) = Heston_Call(100, K+(j-1)*5, 2, rho + 0.3*(i-1),
0.05, 1, 0.4, 0.05, 0, 1)
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  ret = ImVol';
end

function ret = xiMapH(xi, K )
  for i=1:5;
    for j= 1:9;
      price(i,j) = Heston_Call(100, K+(j-1)*5, 2, -0.7, 0.05, 1, xi
+ (i-1)*0.2, 0.05, 0, 1);
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  ret = ImVol';
end

function ret = thetaMap(theta, K )
  for i=1:5;
    for j= 1:9;
      price(i,j) = Heston_Call(100, K+(j-1)*5,2, -0.7, theta + 0.1*(i-1),
1, 0.4, 0.05, 0, 1) ;
      ImVol(i,j)=blsimpv(100, K+(j-1)*5, 0,1,price(i,j));
    end
  end
  ret = ImVol';
end
```

A.5 SABR Calibration Scheme

This section contains the code for calibrating the SABR model over the time period (i.e. it is the code for the time series of parameters).

If one wants to calibrate in the case where β and ρ are fixed, one simply comments out the sections of the code used for the case in which β only is fixed (see the comments in the code).

A sample of the data used has been included to aid in the reading and understanding of the code.

A.5.1 Sample of spreadsheet data

Strike	Forward	Size	Vol	Term	B/S	AtmVol	C/P	dft	ID	Trade	flag	TT			EF	Efl	premium	RD
26000	27391.77	2500	0.246	1.1753	-1	0.240699	-1	506	160000126	12-Oct-10	1	1	-	-	1	1	2189.697	412
26000	27559.86	500	0.250	1.1315	1	0.235769	-1	490	160000159	28-Oct-10	2	1	-	-	2	1	2120.127	396
27000	27782.15	500	0.273	1.0411	1	0.242273	-1	457	160000206	30-Nov-10	3	1	-	-	3	1	2655.383	363
26000	28565.42	1000	0.266	1.0247	1	0.249234	-1	451	160000214	06-Dec-10	4	6	-	-	4	6	1814.4	357
29000	28565.42	1750	0.238	1.0247	-1	0.249234	-1	451	160000214	06-Dec-10	4	5	-	-	4	5	2977.879	357
32000	28565.42	1000	0.221	1.0247	1	0.249234	1	451	160000214	06-Dec-10	4	4	-	-	4	4	1321.8	357
26000	28626.78	2000	0.264	1.0247	1	0.249234	-1	451	160000214	06-Dec-10	4	3	-	-	4	3	1772.6	357
29000	28626.78	3500	0.235	1.0247	-1	0.249234	-1	451	160000214	06-Dec-10	4	2	-	-	4	2	2920.346	357
32000	28626.78	2000	0.222	1.0247	1	0.249234	1	451	160000214	06-Dec-10	4	1	-	-	4	1	1347.6	357
26000	29140.44	2000	0.270	1.0219	-1	0.249594	-1	450	160000215	07-Dec-10	5	2	-	-	5	2	1680.7	356
26000	29145.56	500	0.270	1.0219	-1	0.249594	-1	450	160000215	07-Dec-10	5	1	-	-	5	1	1680.7	356
25000	28845.31	500	0.286	1.0164	1	0.248824	-1	448	160000218	09-Dec-10	6	1	-	-	6	1	1533	354
25000	28720.07	500	0.279	1.0055	1	0.248506	-1	444	160000221	13-Dec-10	7	1	-	-	7	1	1484.9	350

Figure 37: Sample of data spreadsheet used in Heston calibration

Figure 14 shows a sample of the spreadsheet used as data in the calibrator. The first eleven columns are the same as those shown and explained in Table 1, §2.2. The remaining variables have been inserted for use in the calibrator.
"flag" denotes the number (chronologically) of the package in which an option is contained. The last entry in the dataset denotes the number of packages in the dataset (in the sample, it is 5).
"TT" denotes what number (in reverse order) the option in a package is (for example, on 6 Dec 2010, the package contains 6 options: the first option in the package is given a 6, the second a 5 and so forth).
"spot" is the price of the underlying on the trade date given.
"-" is a placeholder column.
"EF" denotes the trade day on which an option occurs.
"EFl" denotes the number (in reverse order) that the option is on a trade-day (for example, on 6 Dec 2010, 6 options were sold and the first option sold is given a 6, the second a 5 and so forth).
"Premium" is the price of the option under the Black-Scholes model.
"i" is the interest rate on that trade-day.

"RD" denotes the number of days necessary to roll from one day's calibration to the next.

A.5.2 SABR Calibration Scheme

```
clear;
load Thesis_data.mat;
global data;
global NoOfOptions;
global NoOfIterations;
global NoOfPackages;
global datanew;
global tradedays;
global beta

beta = 0.7;
NoOfIterations = 0;
Size = size(data);
NoOfOptions = Size(1);
tradedays = data(NoOfOptions, 16);
k = 24;
result = zeros(tradedays-13, 2);

%x0 = [ -0.75 0.2];
%lb = [ -0.99 0.01];
%ub = [ -0.02 4];

for i = 1 :  tradedays-13
  datanew = data(1:k, :);
  n = datanew(k, 17);
  Size = size(datanew);
  NoOfOptions = Size(1);
  NoOfPackages = datanew(NoOfOptions, 12)
  options = optimset('MaxFunEvals',20000, 'TolFun', 1e-08, 'TolX',
1e-08); %sets the max no. of iteration to 20000 so that termination doesn't
take place early. Change this number if you find it is taking too long.
  tic;
    Calibration = lsqnonlin(@SABR_ObjFunc_Evo, x0, lb, ub);
  toc;
  Solution = [Calibration(1), Calibration(2)]
  k = k + n;
  result(i,:)  = Solution
end
ret = result;

function y = SABR_ObjFunc_Evo(input)
```

```
global NoOfOptions;
global data;
global NoOfIterations;
global pricedifference;
global beta
global NoOfPackages;

NoOfIterations = NoOfIterations + 1;
k = 1

for j = 1 :  NoOfPackages
  n = data(k, 13);
  PricePackage(j) = data(k,6)*data(k, 3)*(Black76(data(k, 2),data(k,1),...
  data(k,5), data(k,4), data(k,8)));
  a = FindAlpha(data(k,2),data(k,1), data(k,5), data(k,7), beta, input(1),
input(2));
  SABR_vol(j) = SABRvol(a,beta,input(1),input(2),data(k,2),data(k,1),data(k,5));

  SABRPackage(j) = data(k,6)*data(k,3)*Black76(data(k,2), data(k,1),
data(k,5), ...
  SABR_vol(j), data(k,8));
  for i = 1 :  n-1
    PricePackage(j) = PricePackage (j) + data(k+i, 3)*data(k+i, 6)*...
    (Black76(data(k+i, 2),data(k+i,1), data(k+i,5), data(k+i,4), data(k+i,8)));
    a = FindAlpha(data(k+i,2),data(k+i,1), data(k+i,5), data(k+i,7),
    ...beta, input(1), input(2));
    SABR_vol(k+i) = SABRvol(a,beta,input(1),input(2),data(k+i,2),...
    data(k+i,1),data(k+i,5));
  end
 relativeDiff(j) = (PricePackage(j) - SABRPackage(j))/PricePackage(j);
  pricedifference(j) = 0.99^ (data(k, 9)- 87 - rollday)*relativeDiff(j)*relativeDiff(j);

  k = k+n;
end

y = pricedifference';
end
```

A.6 Heston Calibration Scheme

This section contains the Heston calibrator. The code is for the calibration over a period of time (i.e. recalibrations on every day a trade occurs, over a period of a year). The script is similar to the SABR calibrator with the only differences being the specified bounds, the number of inputs and the objective function.

If one wants to calibrate in the case where κ (kappa) and θ (theta) are fixed, one simply comments out the sections of the code used for the full calibration (see the comments in the code).

```
clear;
load Thesis_data.mat;
global data;
global NoOfOptions;
global NoOfIterations;
global NoOfPackages;
global datanew;
global tradedays;
global kappa;
global theta;

%kappa = 2;        %Uncomment these two lines in the case where kappa and
theta are fixed and calibration is of the remaining three parameters.
%theta = 0.05;
NoOfIterations = 0;
Size = size(data);
NoOfOptions = Size(1);
tradedays = data(NoOfOptions, 16);
k = 24;
result = zeros(tradedays-13, 3);

%x0 = [ 0.8 0.23 -0.75];
%lb = [ 0.2 0.02 -0.99];
%ub = [ 4 0.4 -0.2];
x0 = [ 0.8 0.23 -0.75 3.5 0.23];        %Use these lines of code in place
of the above three lines of code when fixing parameters.
lb = [ 0.2 0.02 -0.99 0 0.02];
ub = [ 4 0.4 -0.2 8 0.6];

for i = 1 :  tradedays-13
  datanew = data(1:k, :);
  n = datanew(k, 17);
  Size = size(datanew);
  NoOfOptions = Size(1);
  NoOfPackages = datanew(NoOfOptions, 12)
  options = optimset('MaxFunEvals',20000, 'TolFun', 1e-08, 'TolX',
1e-08); %sets the max no. of iterations to 20000 so that termination doesn't
take place early. Change this number if you find it's taking too long.
  tic;
    Calibration = lsqnonlin(@Heston_ObjFunc_Evo, x0, lb, ub);
  toc;
```

```
  Solution = [Calibration(1), Calibration(2), Calibration(3)], Calibration(4),
Calibration(5)];
%Solution = [Calibration(1), Calibration(2), Calibration(3)]      %For
calibration with fixed parameters
  k = k + n;
  result(i,:)  = Solution
end
ret = result
```

Note, in the objective function, in the case where the parameters are fixed, `input(4)` will be replaced with `kappa` and `input(5)` with be replaced with `theta`.

```
function y = Heston_ObjFunc_Evo(input)

global NoOfOptions;
global data;
global NoOfIterations;
global pricedifference;
global kappa;
global theta;
global NoOfPackages;

NoOfIterations = NoOfIterations + 1;
k = 1

for j = 1 :  NoOfPackages
  n = data(k, 13);
  PricePackage(j) = datadata(k, 18)*data(k,3) ;
  HestonPackage(j)=data(k,3)*Heston_Call(data(k,2),data(k,1),...
input(4), input(3),input(5), data(k,5), input(1),input(2),...
data(k,19), data(k, 8));

  for i = 1 :  n-1
    PricePackage(j) = PricePackage (j) + data(k+i,3)*data(k+i, 18);

    HestonPackage(j)=HestonPackage(j)+data(k+i,3)*data(k+i,6)...
*Heston_Call(data(k+i,2),data(k+i,1),input(4),input(3),input(5),...
data(k+i,5),input(1), input(2), data(k+i,19), data(k+i, 8));
  end
 relativeDiff(j) = (PricePackage(j) - HestonPackage(j))/PricePackage(j);
  pricedifference(j) = 0.99^ (data(k, 9)- 87 - rollday)*relativeDiff(j)*relativeDiff(j);

  k = k+n;
end
```

```
y = pricedifference';
end
```

A.7 Errors

This section contains the code for extracting the errors of the Heston model. An excel file "params.xlsx" contains the vectors of parameters output by the calibration over time.

```
clear;
Parameter = xlsread('params.xlsx');
global datanew;
global NoOfIterations;
global NoOfPackages;
global tradedays;
global data;
global rollday;

load Thesis_data.mat;

NoOfIterations = 0;
Size = size(data);
NoOfOptions = Size(1);
tradedays = data(NoOfOptions, 16);
Solution = zeros(1, 10);

k = 24;
for i = 1 :  tradedays-13
  kappa = Parameter(i, 4);
  theta = Parameter(i, 5);
  xi = Parameter(i, 1);
  Vo = Parameter(i, 2);
  rho = Parameter(i, 3);
  datanew = data(1:k, :);
  Size = size(datanew);
  NoOfOptions = Size(1);
  NoOfPackages = datanew(NoOfOptions, 12);
  n = data(k, 17);
  rollday = data(k, 23);

  Solution = ErrorCalc(xi, Vo, rho, kappa, theta);
  SolSize = size(Solution);
  Sizewewant = SolSize(2);
  for j = 1:  Sizewewant
    Error(i,j) = Solution(j);
```

```
  end
  k = k + n;
end

function y = ErrorCalc(xi, Vo, rho, kappa, theta, k)
  global NoOfPackages;
  global data;
  global rollday;

  k = 1 ;
  for j = 1:NoOfPackages
    n = data(k, 13);
    PricePackage(j) = data(k,3)*data(k,6)*data(k, 18);
    HestonPackage(j) = data(k,3)*data(k,6)*Heston_Call_adj(data(k,2),
data(k,1), kappa,
    rho,theta, data(k,5), xi,Vo,     data(k,19), data(k, 8));
    for i = 1 :  n-1
      PricePackage(j) = PricePackage (j) + data(k+i,3)*data(k+i,6)*data(k+i,
18);
      HestonPackage(j)=HestonPackage(j)+data(k+i,3)*data(k+i,6)*Heston_Call_adj(data(k+i,2)
      ,data(k+i,1),kappa,rho,theta,data(k+i,5),xi, Vo, data(k+i,19),
data(k+i, 8));
  end
  relDiff(j) = (PricePackage(j) - HestonPackage(j))/PricePackage(j);

  pricedifference(j) = 0.99^ (data(k, 9)- 87 - rollday)*relDiff(j)*relDiff(j);
  k = k + n;
  end
  y = pricedifference';
end
```

References

Gauthier, P., and P.H. Rivaille. Fitting the Smile: Smart Parameters for SABR and Heston. Available at SSRN, 2009. URL http://ssrn.com/abstract=1496982 or http://dx.doi.org/10.2139/ssrn.1496982

Gilli, M. and E. Schumann. Calibrating Option Pricing Models with Heuristics. Available at SSRN, 2010. URL http://ssrn.com/abstract=1566975

Hagan, P., D. Kumar, A. Lesniewski and D. Woodward. Managing Smile Risk. *WILMOTT magazine*, Sept: 84-108 , 2002.

Heston, S. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies*, 6(2): 327-343, 1993.

Hull, J. *Options, Futures and other Derivative Securities*. Prentice-Hall, seventh edition, 2008.

Kahl, C., and P. Jackel. Not-so-Complex Logarithms in the Heston Model. *WILMOTT magazine*, 19: 94-103, 2005.

Mikhailov, S., and N. Ulrich. Heston's Stochastic Volatility Model Implementation, Calibration and some Extensions. *WILMOTT magazine*, (4): 74-79, 1993.

Moodley, N. The Heston Model: A Practical Approach with Matlab Code. Honours Project, Faculty of Science, The University of Witwatersrand, 2005. URL http://math.nyu.edu/~atm262/fall06/compmethods/a1/nimalinmoodley.pdf

West, G. Calibration of the SABR model in illiquid markets. *Applied Mathematical Finance*, 12(4): 371-385, 2005.

Wilmott, P. Bankers can't avoid Risk by Hiding it. Online website, 2011. URL http://www.wilmott.com/blogs/paul/index.cfm/2011/11/9/Bankers-Cant-Avoid-Risk-by-Hiding-It