Time Consistency in Option Pricing Models

Johan Nykvist

October 30, 2009

Abstract

Since the introduction of the famous Black-Scholes model (1973), several attempts have been made to construct option pricing models that allow for non-gaussian return distributions as well as varying volatilities. In this thesis, we examine the robustness of two of these models in terms of the time consistency, or possibly inconsistency, of the model parameters. We restrict our attention to the stochastic volatility model provided by Heston (1993) and the local volatility model introduced by Dupire (1994). We estimate the models daily in order to find parameters that match the current market prices as closely as possible, hence the calibration process constitutes a major part of the thesis. Our results show that both models are succesful in explaining important characteristics of the implied volatility surface, when the market conditions are fairly stable. On the other hand, when the market is heavily fluctuating, both models reveal a high degree of time inconsistency, as they are unable to capture the current market conditions without large parameter variations. In addition, the use of principal component analysis shows that variations of the local volatility surface, to a large extent can be explained by three distinct movements.

Acknowledgements

I would like to thank my tutor Filip Lindskog for helpful advice during the writing of this thesis. I am also grateful to Jonas Kiessling for valuable suggestions.

Stockholm, October 2009

Johan Nykvist

Contents

1 Introduction 1

2 Model Introduction and Data 3

2.1 The Heston Model: Stochastic volatility 3

2.2 Heston Continued 6

2.3 The Dupire Model - Local Volatility 9

2.4 Data description 11

3 The Calibration Problem 13

3.1 Calibrating Heston in Theory 13

3.2 Calibrating Heston in Practice 15

3.3 Calibrating Dupire 18

4 Calibration Results 23

4.1 The Heston Model 23

4.2 The Dupire Model 27

5 Conclusions 33

A Results 37

A.1 Principal Component Analyis 37

B Matlab code 39

B.1 Calibrating the Heston model using Differential Evolution . . 39

B.2 Calibrating the Heston model using *lsqnonlin* 42

B.3 Calibrating the Dupire model using *lsqnonlin* 44

Chapter 1

Introduction

Since the introduction of the famous Black-Scholes model (1973), several attempts have been made to construct alternative option pricing models that allow for non-gaussian return distributions as well as non-constant volatility. Models that allow negative correlation between the underlying stock price and its volatility are examples of such models that are commonly found in the literature.

The development of more sophisticated models however comes at the cost of increased complexity. While the Black-Scholes model only have one unknown parameter, stochastic volatility models typically have between five and fifteen parameters that have to be estimated. Consequently, the calibration of such models is in general far more troublesome than calibrating the original model proposed by Black and Scholes.

The performance of stochastic volatility models, in terms of pricing and hedging performance, has been investigated in a large number of papers (see for example Bakshi, Cao and Chen (1997), Christoffersen, Heston and Jacobs (2009) or Shoutens, Simons and Tistaert (2003)). What remains unexamined however is the time consistency, or possibly inconsistency, of these models in terms of parameter variations over time. From a theoretical point of view, a robust model would allow fluctuations in the underlying asset with the parameters remaining fairly constant. Large variations in daily parameter estimates would reduce the usefulness of these models as daily recalibrations are both time consuming and may lead to large daily variations in more exotic contracts (Schoutens, Simons and Tistaert (2003)).

In this thesis we aim to investigate the time consistency of alternative option pricing models. We restrict our attention to the single factor stochastic volatility model proposed by Heston (1993) and the local volatility function introduced by Dupire (1994). The models are applied to a number of put and call options written on the Euro Stoxx 50 index between June 23^{rd} and December 31^{st} 2008.

In Chapter 2 we introduce the two models and provide the theoretical framework needed to proceed with the calibration problem. Since calibration of stochastic volatility models is quite complex, a thorough discussion of different approaches is provided in Chapter 3. In Chapter 4 we present the main results from the empirical study and discuss some of the implications on option pricing theory. In the last chapter we summarize the material covered in the thesis and suggest topics for further studies.

Chapter 2

Model Introduction and Data

2.1 The Heston Model: Stochastic volatility

Assume that the spot price follows the diffusion

$$dS(t) = \mu S(t)dt + \sqrt{v(t)}S(t)dW_1(t), \quad S(0) = S_0,$$

where $W_1(t)$ is a Wiener process. If the volatility follows an Ornstein-Uhlenbeck process,

$$d\sqrt{v(t)} = -\beta\sqrt{v(t)}dt + \delta dW_2(t),$$

where $W_2(t)$ has correlation ρ with $W_1(t)$, then Itô's lemma shows that the variance $v(t)$ follows the process,

$$dv(t) = [\delta^2 - 2\beta v(t)]dt + 2\delta\sqrt{v(t)}dW_2(t), \quad V(0) = V_0.$$

This process may be rewritten as

$$dv(t) = \kappa[\theta - v(t)]dt + \sigma\sqrt{v(t)}dW_2(t),$$

which is known as a square root mean reverting process, first used by Cox, Ingersoll and Ross (1985), with long-run mean θ, and rate of reversion κ. σ is referred to as the volatility of the volatility.

For $\kappa, \theta > 0$, this corresponds to a process where the randomly moving volatility is elastically pulled toward a long-term value, θ. The parameter κ determines the speed of adjustment. In addition, if $2\kappa\theta \geq 0$, the volatility process is always larger than zero (see Cox, Ingersoll and Ross (1985) or Feller (1951)).

Earlier research shows that there are both economical as well as empirical reasons for a model of this structure. Firstly, implied Black-Scholes volatilities vary both with time to maturity and with strike price, so modeling volatility as a random variable is a rather natural approach. Secondly, empirical studies indicate that an asset's log-return distribution is non-Gaussian, characterized by heavy tails. This behavior is captured in the Heston model by the parameter ρ. Intuitively, if $\rho > 0$, the volatility will increase as the asset return increases. This will spread the right tail and squeeze the left tail of the distribution creating a distribution with a fat right tail. The opposite is of course true. In fact, there is evidence that the correlation between asset returns and implied volatility is negative, also known as the 'leverage effect'. ρ, therefore, affects the skewness of the distribution. Finally, a phenomena known as 'volatility clustering' has been readily observed in the market. Basically, it means that large price variations are more likely to be followed by large price variations and vice versa. In the Heston model, the mean reversion parameter κ can also be interpreted as representing the degree of 'volatility clustering'.

Thus, there are several economical as well as theoretical arguments for this choice of model. The main advantage of the Heston model, however, is the closed-form solution for European call options. In the next section, we derive the general valuation equation and apply it to the Heston model in order to obtain a pricing formula for European calls.

2.1.1 Derivation of the Valuation Equation

In this section, we follow the work of Gatheral (2006) closely. We begin by assuming that the spot price and the volatility follow the diffusions

$$dS(t) = \mu S(t)dt + \sqrt{v(t)}S(t)dW_1(t), \tag{2.1}$$

$$dv(t) = \alpha(S(t), v(t), t)dt + \eta\beta(S(t), v(t), t)\sqrt{v(t)}dW_2(t), \tag{2.2}$$

where

$$\langle dW_1(t), dW_2(t)\rangle = \rho dt.$$

In contrast with the Black-Scholes model there are two sources of randomness, the stock price and the volatility. Thus, in order to form a riskless portfolio we set up a portfolio Π containing the option being priced, whose value we denote by $V(S, v, t)$, a quantity $-\Delta$ of the stock and a quantity $-\Delta_1$ of another asset whose value V_1 depends on volatility. We have

$$\Pi = V - \Delta S - \Delta_1 V_1.$$

For the next step we need the following proposition (see Björk (2004), p. 56):

Proposition 2.1. *Take a vector Wiener process $W = (W_1, ..., W_n)$ with correlation matrix ρ as given, and assume that the vector process $X = (X_1, ..., X_k)$ has a stochastic differential. Then the following hold:*

For any $C^{1,2}$ function f, the stochastic differential of the process $f(t, X(t))$ is given by

$$df(t, X(t)) = \frac{\partial f}{\partial t}dt + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}dX_i + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}dX_i dX_j,$$

with the formal multiplication table

$$\begin{cases} (dt)^2 = 0, \\ dt \cdot dW_i = 0, i = 1, ..., n, \\ dW_i \cdot dW_j = \rho_{ij} dt. \end{cases}$$

Applying this proposition shows that the change in Π in a time dt is given by

$$\begin{aligned} d\Pi = &\left\{ \frac{\partial V}{\partial t}dt + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\eta v\beta S\frac{\partial^2 V}{\partial v \partial S} + \frac{1}{2}\eta^2 v\beta^2\frac{\partial^2 V}{\partial v^2} \right\} dt \\ &- \Delta_1 \left\{ \frac{\partial V_1}{\partial t} + \frac{1}{2}vS^2\frac{\partial V_1}{\partial S^2} + \rho\eta v\beta S\frac{\partial^2 V_1}{\partial v \partial S} + \frac{1}{2}\eta^2\beta^2 v\frac{\partial^2 V_1}{\partial v^2} \right\} dt \\ &+ \left\{ \frac{\partial V}{\partial S} - \Delta_1\frac{\partial V_1}{\partial S} - \Delta \right\} dS \\ &+ \left\{ \frac{\partial V}{\partial v} - \Delta_1\frac{\partial V_1}{\partial v} \right\} dv \end{aligned}$$

To make the portfolio instantaneously risk-free, we must choose

$$\begin{aligned} &\frac{\partial V}{\partial S} - \Delta_1\frac{\partial V_1}{\partial S} - \Delta = 0 \\ &\frac{\partial V}{\partial v} - \Delta_1\frac{\partial V_1}{\partial v} = 0 \end{aligned}$$

to eliminate dS terms and dv terms respectively. This leaves us with

$$\begin{aligned} d\Pi = &\left\{ \frac{\partial V}{\partial t}dt + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\eta v\beta S\frac{\partial^2 V}{\partial v \partial S} + \frac{1}{2}\eta^2 v\beta^2\frac{\partial^2 V}{\partial v^2} \right\} dt \\ &- \Delta_1 \left\{ \frac{\partial V_1}{\partial t} + \frac{1}{2}vS^2\frac{\partial V_1}{\partial S^2} + \rho\eta v\beta S\frac{\partial^2 V_1}{\partial v \partial S} + \frac{1}{2}\eta^2\beta^2 v\frac{\partial^2 V_1}{\partial v^2} \right\} dt \\ &= r\Pi dt \\ &= r(V - \Delta S - \Delta_1 V_1)dt, \end{aligned}$$

where we have used the fact that the return on a risk-free portfolio must equal the risk-free rate r. Collecting all V terms on the left-hand side and all V_1 terms on the right-hand side, we get

$$\frac{\frac{\partial V}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2}+\rho\eta v\beta S\frac{\partial V}{\partial v\partial S}+\frac{1}{2}\eta^2 v\beta^2\frac{\partial^2 V}{\partial v^2}+rS\frac{\partial V}{\partial S}-rV}{\frac{\partial V}{\partial v}}$$
$$=\frac{\frac{\partial V_1}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 V_1}{\partial S^2}+\rho\eta v\beta S\frac{\partial V_1}{\partial v\partial S}+\frac{1}{2}\eta^2 v\beta^2\frac{\partial^2 V_1}{\partial v^2}+rS\frac{\partial V_1}{\partial S}-rV_1}{\frac{\partial V_1}{\partial v}}.$$

The left-hand side is a function of V only and the right-hand side is a function of V_1 only. Thus, we conclude that both sides must be equal to some function f of the *independent* variables S, v and t. We deduce that

$$\begin{aligned}&\frac{\partial V}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2}+\rho\eta v\beta S\frac{\partial V}{\partial v\partial S}+\frac{1}{2}\eta^2 v\beta^2\frac{\partial^2 V}{\partial v^2}+rS\frac{\partial V}{\partial S}-rV\\&=-\left(\alpha-\phi\beta\sqrt{v}\right)\frac{\partial V}{\partial v}\end{aligned}\tag{2.3}$$

where, without loss of generality, we have written the arbitrary function f of S, v and t as $(\alpha-\phi\beta\sqrt{v})\frac{\partial V}{\partial v}$. $\phi(S, v, t)$ is called the market price of volatility risk. Now, defining the *risk-neutral drift* as

$$\alpha' = \alpha - \beta\sqrt{v}dZ_2$$

we see that, as far as pricing of options is concerned, we could have started with the risk-neutral SDE for v,

$$dv = \alpha' dt + \beta\sqrt{v}dZ_2 \tag{2.4}$$

and got identical results with no explicit price of risk term because we are in the risk-neutral world. In what follows, we assume that the SDEs for S and v are in risk-neutral terms because we are invariably interested in fitting models to option prices.

2.2 Heston Continued

The Heston model corresponds to choosing $\alpha(S(t), v(t), t) = \kappa(\theta - v(t))$ and $\beta(S(t), v(t), t) = 1$ in equations (2.1) and (2.2). These equations then become

$$dS(t) = \mu S(t)dt + \sqrt{v(t)}S(t)dW_1(t)$$

and

$$dv(t) = \kappa(\theta - v(t))dt + \sigma\sqrt{v(t)}dW_2(t)$$

with

$$\langle dW_1(t), dW_2(t)\rangle = \rho dt$$

where κ is the speed of reversion of $v(t)$ to its long-term mean θ. We now substitute the above values for $\alpha(S(t), v(t), t)$ and $\beta(S(t), v(t), t)$ into the general valuation equation. We obtain

$$\begin{aligned}\frac{\partial V}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\sigma vS\frac{\partial^2 V}{\partial v \partial S} + \frac{1}{2}\sigma^2 v\frac{\partial^2 V}{\partial v^2} + rS\frac{\partial V}{\partial S} - rV& \\ = -\kappa(\theta - v)\frac{\partial V}{\partial v}& \qquad (2.5)\end{aligned}$$

In Heston's original paper, the price of risk is assumed to be linear in the variance v. In contrast, we assume that the Heston process, with parameters fitted to option prices, generates the risk-neutral measure so the market price of volatility risk ϕ in the general valuation equation (2.3) is set to zero. Since we are only interested in pricing, and we assume that the pricing measure is recoverable from European option prices, we are indifferent to the statistical measure.

2.2.1 The Heston Solution for European Options

A European call option with strike price K and maturing at time T satisfies the partial differential equation (2.5) subject to the following boundary conditions:

$$\begin{aligned}V(S, v, T) &= \max(S - K, 0),\\ V(0, v, t) &= 0,\\ \frac{\partial V}{\partial S}(\infty, v, t) &= 1, \qquad (2.6)\\ rS\frac{\partial V}{\partial S}(S, 0, t) + \kappa\theta\frac{\partial V}{\partial v}(S, 0, t) - rV(S, 0, t) + \frac{\partial V}{\partial t}(S, 0, t) &= 0,\\ V(S, \infty, t) &= S\end{aligned}$$

In analogy with the Black-Scholes formula, for the European call option, whose value we denote $C(S, v, t)$ we guess a solution of the form

$$C(S, v, t) = SP_1 - KP(t, T)P_2, \qquad (2.7)$$

where the first term is the present value of the spot price upon optimal exercise, and the second term is the present value of the strike-price payment. $P(t, T)$ is the price at time t of a zero-coupon bond maturing at time T with face value 1. Both of these terms must satisfy the original PDE (2.5). If we define $x = \ln S$ and substitute the proposed solution (2.7) into the original PDE (2.5) we see that P_1 and P_2 must satisfy the PDEs

$$\begin{aligned}&\frac{1}{2}v\frac{\partial^2 P_j}{\partial x^2} + \rho\sigma v\frac{\partial^2 P_j}{\partial x \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 P_j}{\partial v^2} + (r + u_j v)\frac{\partial P_j}{\partial x}\\ &+ (a_j - b_j v)\frac{\partial P_j}{\partial v} + \frac{\partial P_j}{\partial t} = 0, \qquad (2.8)\end{aligned}$$

for $j = 1, 2$, where

$$u_1 = 1/2, u_2 = -1/2, a = \kappa\theta, b_1 = \kappa - \rho\sigma, b_2 = \kappa.$$

Given the boundary conditions for the option price in equation (2.6), these PDEs (2.8) are subject to the terminal condition

$$P_j(x, v, T; \ln K) = 1_{\{x \geq \ln K\}}.$$

Thus, they may be interpreted as "adjusted" or "risk-neutralized" probabilities (see Cox and Ross (1976)). To see why, assume that $x(t)$ and $v(t)$ follows the stochastic process

$$\begin{aligned} dx(t) &= (r + u_j v)dt + \sqrt{v(t)}dz_1(t), \\ dv(t) &= (a_j - b_j v)dt + \sigma\sqrt{v(t)}dz_2(t), \end{aligned}$$

where the parameters u_j, a_j, and b_j are defined as before. Further, consider any twice-differentiable function $f(x, v, t)$ that is a conditional expectation of some function of x and v at a later date, T, $g(x(T), v(T))$:

$$f(x, v, t) = E[g(x(T), v(T))|x(t) = x, v(t) = v]. \tag{2.9}$$

Using Itô's lemma we get

$$\begin{aligned} df = &\left(\frac{1}{2}v\frac{\partial^2 f}{\partial x^2} + \rho\sigma v\frac{\partial^2 f}{\partial x \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 f}{\partial v^2} + (r + u_j v)\frac{\partial f}{\partial x} + (a - b_j v)\frac{\partial f}{\partial v} + \frac{\partial f}{\partial t}\right) dt \\ &+ (r + u_j v)\frac{\partial f}{\partial x}dz_1 + (a - b_j v)\frac{\partial f}{\partial v}dz_2. \end{aligned} \tag{2.10}$$

By iterated expectations, we know that f must be a martingale, in particular, $E[df] = 0$. Applying this to equation (2.10) yields the well-known Fokker-Planck forward equation:

$$\begin{aligned} &\frac{1}{2}v\frac{\partial^2 f}{\partial x^2} + \rho\sigma v\frac{\partial^2 f}{\partial x \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 f}{\partial v^2} \\ &\qquad (r + u_j v)\frac{\partial f}{\partial x} + (a - b_j v)\frac{\partial f}{\partial v} + \frac{\partial f}{\partial t} = 0 \end{aligned}$$

Equation (2.9) imposes the terminal condition

$$f(x, v, T) = g(x, v).$$

Now, if $g(x, v) = 1_{\{x \geq \ln K\}}$, then the solution is the conditional probability at time t that $x(T)$ is greater than $\ln K$. In addition, if $g(x, v) = e^{i\phi x}$, then the solution is the characteristic function $E[e^{i\phi x(T)}|x(t) = x, v(t) = v]$. The probabilities are not immediately available in closed form. However, Heston shows that their characteristic functions, $f_1(x, v, T; \phi)$ and $f_2(x, v, T, \phi)$ respectively, satisfy the same PDEs (2.8), subject to the terminal condition

$$f_j(x, v, T; \phi) = e^{i\phi x}.$$

The characteristic function solution is

$$f_j(x, v, T; \phi) = e^{C(T-t;\phi)+D(T-t;\phi)v+i\phi x}, \tag{2.11}$$

where

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

and

$$\begin{aligned} g &= \frac{b_j - \rho\sigma\phi i + d}{b_j - \rho\sigma\phi i - d}, \\ d &= \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2(2u_j\phi i - \phi^2)}. \end{aligned}$$

One can invert the characteristic functions to get the desired probabilities:

$$P_j(x, v, T; \ln K) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty Re\left[\frac{e^{-i\phi\ln K}f_j(x, v, t; \phi)}{i\phi}d\phi\right]. \tag{2.12}$$

The integrand in equation (2.12) is a smooth function that decays rapidly. Equations (2.7), (2.11), and (2.12) give the solution for European call options.

2.3 The Dupire Model - Local Volatility

Given the computational complexity of stochastic volatility models and the difficulty of parameter fitting to plain vanilla options, practitioners sought a simpler way of pricing options consistently with the volatility skew. The breakthrough came when Dupire (1994) and Derman and Kani (1994) noted that under risk neutrality, there was a unique diffusion process consistent with the risk-neutral density derived from the market prices of European options. The corresponding unique diffusion coefficient $\sigma_L(S, t)$ is known as the *local volatility function.* In the next section we review the original work of Dupire and derive an explicit expression for the local volatility function.

2.3.1 Derivation of the Dupire Equation

Suppose the stock price diffuses with risk-neutral drift $\mu(t) = r(t) - D(t)$ and local volatility $\sigma(S, t)$ according to the equation

$$\frac{dS(t)}{S(t)} = \mu(t)dt + \sigma(S(t), t)dW(t),$$

where $r(t)$ is the risk-free interest rate, $D(t)$ is the dividend yield and $W(t)$ is a Wiener process. The risk-neutral expected value $C^*(S_0, K, T)$ of a European option payoff with strike K and expiration T is given by

$$C^*(S_0, K, T) = \int_0^{\infty} \varphi(S_T, T; S_0)(S_T - K)^+ dS_T, \tag{2.13}$$

where $\varphi(S_T, T; S_0)$ is the probability density of the final spot at time T. For a process with drift $D_1(x,t)$ and diffusion $D_2(x,t)$ the probability density function evolves over time according to the Fokker-Planck equation (or Forward Kolmogorov equation) (see Gatheral (2006)):

$$\frac{\partial}{\partial t} = -\frac{\partial}{\partial x}[D_1(x,t)\varphi(x,t)] + \frac{\partial^2}{\partial x^2}[D_2(x,t)\varphi(x,t)].$$

Thus, in the Dupire model the probability density evolves according to the equation

$$\frac{\partial \varphi}{\partial T} = \frac{1}{2}\frac{\partial^2}{\partial S_T^2}\left(\sigma^2 S_T^2 \varphi\right) - \frac{\partial}{\partial S_T}(\mu S_T \varphi)$$

Differentiating (2.13) with respect to K gives

$$\begin{aligned} \frac{\partial C^*}{\partial K} &= -\int_K^{\infty} \varphi(S_T, T; S_0) dS_T \\ \frac{\partial^2 C^*}{\partial K^2} &= \varphi(K, T; S_0) \end{aligned}$$

Now, differentiating (2.13) with respect to time gives

$$\begin{aligned} \frac{\partial C^*}{\partial T} &= \int_K^{\infty} \left\{\frac{\partial}{\partial T}\varphi(S_T, T; S_0)\right\}(S_T - K) dS_T \\ &= \int_K^{\infty} \left\{\frac{1}{2}\frac{\partial^2}{\partial S_T^2}\left(\sigma^2 S_T^2 \varphi\right) - \frac{\partial}{\partial S_T}(\mu S_T \varphi)\right\}(S_T - K) dS_T \end{aligned}$$

Integrating by parts twice gives:

$$\begin{aligned} \frac{\partial C^*}{\partial T} &= \frac{\sigma^2 K^2}{2}\varphi + \int_K^{\infty} \mu S_T \varphi dS_T \\ &= \frac{\sigma^2 K^2}{2}\frac{\partial^2 C^*}{\partial K^2} + \mu(T)\left(-K\frac{\partial C^*}{\partial K}\right), \end{aligned}$$

which is the Dupire equation when the underlying stock has risk-neutral drift μ. Formally, we can solve for the volatility to get

$$\sigma^2(S,t) = \left.\frac{\frac{\partial C^*}{\partial T} + \mu(T) K \frac{\partial C^*}{\partial K}}{\frac{1}{2}K^2 \frac{\partial^2 C^*}{\partial K^2}}\right|_{K=S, T=t} \tag{2.14}$$

The right-hand side of equation (2.14) can be computed from known European option prices. So, given a complete set or European option prices for all strikes and expirations, local volatilities are given uniquely by equation (2.14). Alternatively, assuming that the call price is a function of the implied Black and Scholes volatility, i.e. $C = C_{BS}(S_0, t, T, K, \sigma_I)$, we may calculate the derivatives using the chain-rule and obtain an expression for the local volatility in terms of implied volatilies (see Elder (2002)):

$$\sigma^2(S,T) = \left. \frac{\sigma_I^2 + 2\sigma_I(T-t)\left(\frac{\partial \sigma_I}{\partial T} + (r-D)K\frac{\partial \sigma_I}{\partial K}\right)}{\left(1 + Kd_1\frac{\partial \sigma_I}{\partial K}\sqrt{T-t}\right)^2 + \sigma_I^2 K^2 (T-t)\left(\frac{\partial^2 \sigma_I}{\partial K^2} - d_1\left(\frac{\partial \sigma_I}{\partial K}\right)^2\sqrt{T-t}\right)} \right|_{K=S,T=t}$$

where

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)(T - t_0)}{\sigma\sqrt{T - t_0}}.$$

Thus, we have established the theoretical framework needed to proceed with the calibration problem. Before that, however, we present the market data used in the calibration process and discuss some of the difficulties with real market data.

2.4 Data description

The data used for the analysis are European style put and call options written on the Euro Stoxx 50 index during the period June 23^{rd} to December 30^{th} 2008. The chosen time period is interesting for a number of reasons. During the first three months, the market is stable with only small fluctuations in the underlying asset. However, the last three months constitute a period of extreme movements on the stock market due to the financial crisis last year. Thus, the robustness of the models in terms of parameter variations will be tested in a period when market conditions vary significantly.

The initial data set consists of four put and twenty call options on the index during the time period specified above. For all options we extract information about maturity, strike price and current index level. The contracts chosen are the most frequently traded options on the index during the time period, making them reliable for the calibration procedure. All in all the set of options consists of contracts with five different strikes (ranging from 2500 to 3500) and four different maturities where, on the first date, the option maturities range from 1 year to 2.5 years. Then, as we monitor these options over time, the maturities decrease continuously, hopefully enabling us to capture the dependence of changes in market expectations on the parameter estimates of the two models.

The yield curve is constructed by a cubic spline interpolation between Euribor quotes of maturities ranging from 1 month to 2 years. All interest rates are assumed to be continuously compounded. The dividend yields are

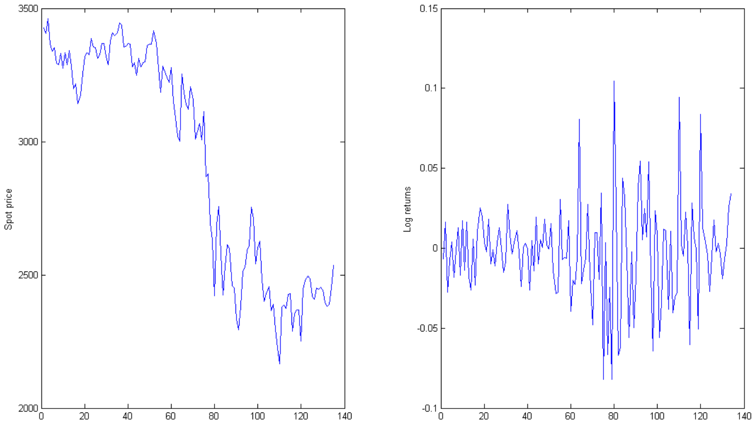

Figure 2.1: The left plot shows the Euro Stoxx 50 index during the sample period and the right plot shows the corresponding log returns. Note the large fluctuations in the spot prices during the last quarter of 2008.

then obtained using the put-call parity (see Section 3.2), where we make use of the fact that on each day we have both put and call prices for the same strike and maturity, which enables us to solve for the implied dividend yield.

Chapter 3

The Calibration Problem

3.1 Calibrating Heston in Theory

We analyze the model in terms of the risk-neutral volatility process described in equation (2.4), because the risk-neutral process exclusively determines prices. Calibrating the Heston model is equivalent to finding the parameters $\kappa, \theta, \sigma, \rho$ and the spot variances V_0 which produce the correct *market prices.* Option pricing models are usually calibrated to market data by minimization of an error functional, i.e. by solving an optimization problem of the form

$$\hat{\Theta} = \arg\min_{\Theta} L[\{C\}^n, \{C(\Theta, \Lambda)\}^n],$$

where Θ is the parameter vector and Λ is the vector of spot variances. $\big\{C(\Theta, \Lambda)\big\}^n$ is a set of n option prices obtained from the model, $\{C\}^n$ is the corresponding set of observed option prices in the market and L is some loss function. From an economic viewpoint, there are several possibilities to measure the error between the market and the model. These different specifications of the error lead to different sets of calibrated model parameters and the resulting pricing performance may vary significantly (Shoutens, Simons and Tistaert (2003)).

3.1.1 The Choice of Loss Function

The most frequently used loss functions in the literature are the dollar mean squared error ($ MSE), the percentage mean squared error (% MSE), and the implied volatility mean squared error (IV MSE):

$$\begin{aligned}
\$\text{MSE}(\Theta,\Lambda) &= \sum_{i=1}^{n} w_i (C_i - C_i(\Theta,\Lambda))^2; \\
\%\text{MSE}(\Theta,\Lambda) &= \sum_{i=1}^{n} w_i \left(\frac{C_i - C_i(\Theta,\Lambda)}{C_i}\right)^2; \\
\text{IV MSE}(\Theta,\Lambda) &= \sum_{i=1}^{n} w_i (\sigma_i - \sigma_i(\Theta,\Lambda))^2,
\end{aligned}$$

where σ_i is the Black-Scholes implied volatility of option i, and $\sigma_i(\Theta,\Lambda)$ denotes the corresponding Black-Scholes implied volatility obtained using the model price as input. w_i is an appropriately chosen weight, which will be discussed in more detail below.

The choice of loss function is important and has many implications. The \$ MSE function minimizes the squared dollar error between model prices and market prices and will thus favor parameters that correctly price expensive options. In contrast, the % MSE function adjusts for price level, and will instead focus on options with prices close to zero. The IV MSE function on the other hand minimizes implied volatility errors, and will therefore favor options with high implied volatilities. Detlefsen and Härdle (2006) has studied four different error functionals and suggest that once a suitable model has been chosen the IV MSE function is best suited for calibrating the model parameters. On the other hand, if there is uncertainty about the correctness of the model, the authors suggests using the \$ MSE function.

Since we are interested in the parameter variations over time, all three loss function will be used during the calibration process. Hopefully, this will lead to a more profound understanding of the dynamics contained in the Heston model.

3.1.2 The Choice of Weights

Earlier research (e.g. Mikhailov and Nögel (2003)) has shown that the choice of weighting w_i has a large influence on the error functional, and therefore on the parameter estimates. Two common methods are to either use the bid-ask spread of the options or to choose weights according to the number of options within different maturity categories. Using $w_i = \frac{1}{|bid_i - ask_i|}$ is a rather intuitive approach. If the bid-ask spread is large, there is a great uncertainty about the true price of the option and we assign it less weight. Since bid and ask prices may be hard to come by, this method is sometimes difficult to use. An alternative approach is to choose weights so that on each day all maturities have the same influence on the objective function. Moreover, the same weight is assigned to all points of the same maturity.

This leads to the weights

$$w_i = \frac{1}{n_{mat} n_{str}^i}$$

where n_{mat} denotes the number of maturities, and n_{str}^i denotes the number of strikes with the same maturity as observation i.

3.1.3 Regularization

In addition to the objective function that is minimized we add a regularization term. Regularization can be necessary for two reasons: Most commonly proposed error functionals may have several global minima (Cont and Hamida (2005)) and thus the regularization term is needed to get a unique minimum. Also, it is important to find parameters on subsequent days that lead to similar prices of exotic options. This is essential for the practical applicability of the model. Moreover, adding a regularization term may provide additional stability to the calibration. In accordance with Mikhailov and Nögel we add a regularization term and minimize the following function

$$L(\{C\}^n, \{C(\Theta, \Lambda)\}^n) + \alpha |\Theta - \Theta_0|,$$

where Θ_0 is the initial parameter vector. The choice of α is important and may have severe implications on the pricing performance of the model. If α is chosen large, the difference between market and model prices has little impact on the total value of the error functional, and over time the pricing performance may decrease. On the other hand, if α is chosen small, the parameters may exhibit rough oscillations leading to large fluctuations in exotic option prices (see Schoutens, Simons and Tistaert (2003)).

3.2 Calibrating Heston in Practice

To calculate model prices in practice we note that the call price is not only a function of the parameters $\kappa, \theta, \sigma, \rho$, and V_0, but also of the strike price K, the time to maturity T, the risk-free interest rate r and the dividend yield δ. The strike price and the time to maturity is uniquely specified by the contract in question, but the risk-free interest rate and the dividend yield have to be approximated in some way. Firstly, we approximate the yield curve using the Euribor. These are given for six different maturities (1 month, 2 months, 3 months, 6 months, 1 year, and 2 years). Then, for any maturity between these values we can find the corresponding interest rate by some suitable interpolation technique. To approximate the dividend yield we assume that the dividends of the Euro Stoxx 50 index are paid continuously. Then, using the next proposition we may solve for the corresponding dividend yield.

Proposition 3.1. (Put-Call Parity when the Underlying pays a Continuous Dividend) *Let $C_\delta(P_\delta)$ denote the price of a European call (put) and let r*

denote the risk-free interest rate, δ denote the dividend yield, S be the price of the underlying, and K be the strike price. Then, the following relation holds:

$$P_\delta = C_\delta - Se^{-\delta(T-t)} + Ke^{-r(T-t)}$$

Proof. Consider a portfolio Π_1 that consists of one long position in a call option and $Ke^{-r(T-t)}$ long positions in zero coupon bonds, i.e.

$$\Pi_1 = C_\delta + Ke^{-r(T-t)}.$$

The value of this portfolio at time T is given by

$$\begin{cases} S - K + K = S, \text{ if } S \geq K \\ 0 + K = K, \text{ if } S \leq K \end{cases}$$

Next, consider a portfolio Π_2 that consists of one long position in a put option and $Se^{-\delta(T-t)}$ long positions in the underlying stock, i.e.

$$\Pi_2 = P + Se^{-\delta(T-t)}$$

The value of this portfolio at time T is given by

$$\begin{cases} 0 + S = S, \text{ if } S \geq K \\ K - S + S = K, \text{ if } S \leq K \end{cases}$$

Since we assume the absence of arbitrage, the present value of the two portfolios must be equal, and this completes the proof. □

Now, for each date we have prices for four call options and four put options on the same underlying asset, with the same strike price and the same time to maturity. Thus, using the proposition above we may solve for the dividend yield to get

$$\delta = \frac{-1}{T-t} \ln \left(\frac{C_\delta - P_\delta + Ke^{-r(T-t)}}{S} \right)$$

Then, we know the risk-neutral drift $\mu(t) = r(t) - \delta(t)$ for four different maturities. Using a smooth cubic spline we interpolate from these values to obtain the drift term for all maturities. Thus, we have obtained all input parameters needed to calculate the model prices. The next step is to solve the optimization problem.

3.2.1 The Optimization Problem

As mentioned earlier, most commonly proposed loss functions are non-convex and may exhibit several local (and perhaps global) minima, making standard optimization techniques unqualified (Cont and Hamida (2005)). Gradient based optimizers, for example, are likely to get stuck in a local minima and are in addition dependent on the initial parameter guess. Thus, option pricing models are often calibrated using some stochastic global optimization routine. Cont and Hamida successfully used an optimization technique called differential evolution when calibrating an option pricing model to real data.

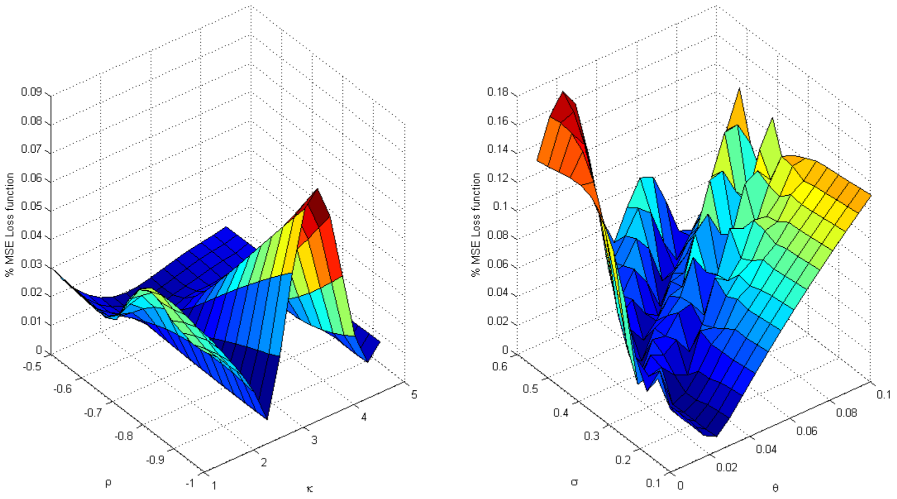

Figure 3.1: The left plot shows the behavior of the % MSE function when κ and ρ are altered. The right plot shows the same loss function when the values of θ and σ are altered.

Consider a search space Θ and a continuous function $G : \Theta \mapsto [0, \infty)$ to be minimized on Θ. An evolutionary algorithm with objective function G is based on the evolution of a population of candidate solutions, denoted by $X_n^N = (\theta_n^i, i = 1, ..., N)$. The basic idea is to 'evolve' the population through cycles of modification (mutation) and selection in order to improve the performance of its individuals. At each iteration the population undergoes three transformations:

$$X_n^N \longmapsto V_n^N \longmapsto W_n^N \longmapsto X_{n+1}^N$$

During the mutation stage, individuals undergo independent random transformations, as if performing independent random walks in Θ, resulting in a randomly modified population V_n^N. In the crossover stage, pairs of individuals are chosen from the population to "reproduce": each pair gives birth to a new individual, which is then added to the population. This new population, W_n^N is now evaluated using the objective function $G(\cdot)$. Elements of the population are now selected for survival according to their fitness: those with a lower value of G have a higher probability of being selected. The N individuals thus selected then form the new population X_{n+1}^N. The role of mutation is to explore the parameter space and the optimization is done through selection.

There is no proof of convergence for algorithms based on differential evolution. However, several comparisons have shown that DE is more accurate and more efficient than several other optimisation techniques including simulated annealing and evolutionary programming. On the downside, stochastic optimization techniques are generally much more time consuming than for example gradient based optimizers. Therefore, we will make use of both alternatives. We use the DE algorithm for the first date to get reliable results. After that we will solve the optimization problem using Matlab's algorithm *lsqnonlin*, which is a gradient based optimizer. On one hand, we risk getting stuck in a local minima. On the other hand, unless the market has changed dramatically we do not expect the parameters to change very much. In periods when the stock price fluctuates heavily we will again use the DE algorithm to obtain reliable parameter estimates.

The Matlab code needed to calibrate the Heston model is included in the Appendix. Despite the fact that the formulae look quite complicated, the calibration of the model is rather straightforward. In this thesis, the integrand in equation (2.12) was calculated using the Matlab function *quadl*. Carr and Madan (1999) present another approach for numerically determining option values, provided that the characteristic function of the risk-neutral density is known. The scheme uses the Fast Fourier Transform (FFT), leading to much faster calculations compared to other numerical integration schemes. Thus, when a large number of options is used in the calibration, the FFT may lead to large reductions in computation time. Because we only used 25 options in the calibration, and also because the Matlab function is more easily implemented, we chose not to use FFT in the calculations.

3.3 Calibrating Dupire

At first glance, the calibration of the local volatility function seems straightforward. Given a complete set of European option prices for all strikes and all expirations, local volatilities are given uniquely by

$$\sigma^2(S,T) = \left. \frac{\frac{\partial C}{\partial T} + \mu(T) K \frac{\partial C}{\partial K}}{\frac{1}{2} K^2 \frac{\partial^2 C}{\partial K^2}} \right|_{K=S, T=t} \tag{3.1}$$

In practice, however, this approach has several shortcomings. Firstly, financial markets typically allow a limited number of prefixed maturity dates, and only a finite number of strikes are on sell, too. Thus, some kind of numerical differentiation method is required to compute the derivatives in the equation above. Secondly, real option prices (for fixed t_0 and S_0) are typically monotonically decreasing in K, and monotonically increasing in T (Hanke and Rösler (2006)). In particular, this implies that the denominator of (3.1) is usually positive. However, positivity of the numerator may not be an obvious property for real data. Therefore, it can easily happen that the fraction changes sign, and taking the square root to obtain σ is prohibited. Finally, interpolation from known data points as well as extrapolation to boundaries outside of the data set may easily result in arbitrage (see Brecher (2006) or Gatheral (2006)).

An alternative approach is to apply the same method used in the calibration of the Heston model parameters. Assume, for now, that given a local volatility function we can calculate the corresponding option prices using the Dupire equation. Then, we can apply the same technique used above and solve an optimization problem of the form

$$\hat{\sigma}(S,T) = \arg\min_{\sigma} L[\{C\}^n, \{C(\sigma)\}^n], \tag{3.2}$$

where, again, $\big\{C(\sigma)\big\}^n$ is a set of n option prices obtained from the model, $\{C\}^n$ is the corresponding set of observed option prices in the market and L is some loss function. Since we already have a working algorithm for solving problems of this form we turn to the problem of calculating option prices using the Dupire model.

3.3.1 Calculating Option Prices using the Dupire Model

Unlike the Heston Model, the local volatility model does not provide a closed-form solution for the prices of plain vanilla options. However, using some finite difference method we can solve the partial differential equation

$$\frac{\partial C}{\partial T} = \frac{1}{2}\sigma^2 K^2 \frac{\partial^2 C}{\partial T^2} - \mu(T) K \frac{\partial C}{\partial K}$$

to obtain prices consistent with the local volatility function. Two different methods are commonly used: the explicit finite difference method and the implicit finite difference method.

3.3.2 The Explicit Finite Difference Method

Let $C_{i,j} = C(i\Delta T, j\Delta K), i,j = 0,...,N$ be the price of a call option with maturity at $T = i\Delta T$ and strike price $K = j\Delta K$. Using a forward difference at time $T = i\Delta T$ we get the recurrence equation

$$\frac{C_{i+1,j} - C_{i,j}}{\Delta T} = \frac{1}{2}\sigma_{i,j}^2 (j\Delta K)^2 \frac{C_{i,j+1} - 2C_{i,j} + C_{i,j-1}}{(\Delta K)^2} - \mu_i j\Delta K \frac{C_{i,j+1} - C_{i,j}}{\Delta K},$$

where μ_i is the drift term at $T = i\Delta T$. Rearranging, we can obtain $C_{i+1,j}$ from the other values by

$$\begin{aligned} C_{i+1,j} =& C_{i,j} + \frac{1}{2}\sigma_{i,j}^2 (j\Delta K)^2 \frac{\Delta T}{(\Delta K)^2}(C_{i,j+1} - 2C_{i,j} + C_{i,j-1}) \\ & - \mu_i (j\Delta K)\frac{\Delta T}{\Delta K}(C_{i,j+1} - C_{i,j}). \end{aligned}$$

Thus, knowing the prices at time i we can obtain the corresponding ones at time $i+1$ using this recurrence relation. However, in order to calculate all prices we need to find suitable boundary conditions. Firstly, we note that

$$C_{o,j} = \max(S - j\Delta K, 0) = (S - j\Delta K)^+,$$

which is the definition of a European call option. This gives us all prices at $T = 0$. In addition, we know that as the strike price approaches infinity, the value of a call option goes towards zero. Thus, if we choose K_N large we might use the approximate boundary condition

$$C_{i,N} = 0.$$

Finally, we note that when $K = 0$ the value of the option will be equal to the price of the underlying for all T, so

$$C_{i,0} = S_0.$$

This explicit method is known to be numerically stable and convergent whenever $\frac{\Delta T}{(\Delta K)^2} \leq \frac{1}{2}$. Thus, to obtain reliable results, we need to interpolate from known data points to a large number of different maturities. Alternatively, we may use the implicit difference method, which always is numerically stable and convergent.

3.3.3 The Implicit Finite Difference Method

If we instead use the backward difference at time $T = (i+1)\Delta T$ we get the recurrence equation

$$\begin{aligned} \frac{C_{i+1,j} - C_{i,j}}{\Delta T} =& \frac{1}{2}\sigma_{i+1,j}^2 (j\Delta K)^2 \frac{C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j}}{(\Delta K)^2} \\ & - \mu_{i+1}(j\Delta K)\frac{C_{i+1,j+1} - C_{i+1,j}}{\Delta K}. \end{aligned}$$

This is an implicit method for solving the Dupire equation (3.1). In each time step we can obtain $C_{i+1,j}$ from solving a system of linear equations

$$\begin{aligned} C_{i,j} =& \left(\mu_{i+1}(j\Delta K)\frac{\Delta T}{\Delta K} - \frac{1}{2}\sigma_{i+1,j}^2 (j\Delta K)^2 \frac{\Delta T}{(\Delta K)^2}\right) C_{i+1,j+1} \\ & + \left(1 + \sigma_{i+1,j}^2 (j\Delta K)^2 \frac{\Delta K}{(\Delta T)^2} - \mu_{i+1}(j\Delta K)\frac{\Delta T}{\Delta K}\right) C_{i+1,j} \\ & - \left(\frac{1}{2}\sigma_{i+1,j}^2 (j\Delta K)^2 \frac{\Delta T}{(\Delta K)^2}\right) C_{i+1,j-1} \end{aligned}$$

subject to the boundary conditions used in the explicit method. The scheme is always numerically stable, but usually more numerically intensive than the explicit method as it requires solving a system of numerical equations on each time step.

3.3.4 Regularization of the Local Volatility Function

In contrast with the Heston model, there is no common practice in regularizing the local volatility function. However, earlier research (e.g. Gatheral (2006) or Brecher (2006)) shows that in order to obtain a smooth volatility surface, some kind of regularization is needed. In this thesis we add a penalty for the curvature of the surface. In particular, when solving the optimization proposed in (3.2) we add two regularization terms and minimize the function

$$\hat{\sigma}(S,T) = \arg\min_{\sigma} \left\{ L[\{C_i\}^n, \{C_i(\sigma)\}^n] + \alpha_K \sum \left| \frac{\partial^2 \sigma}{\partial K^2} \right| + \alpha_T \sum \left| \frac{\partial^2 \sigma}{\partial^2 T} \right| \right\}.$$

The choice of the regularization parameters α_K and α_T are important as they affect both the smoothness of the local volatility function as well as the pricing performance of the model. The second derivatives are calculated numerically using finite differences. The Matlab code needed for the calibration of the Dupire model is included in the Appendix.

3.3.5 Analysing the Dupire Model - Principal Component Analysis

In contrast to the Heston model, the Dupire model does not provide parameters that are easily extracted and interpreted. Instead, every point on the volatility surface is in itself a parameter, which means that, depending on the partition, there are a large number of parameters that have to be analysed. To simplify the analysis we use a variable reduction procedure called *principle component analysis* (PCA). PCA is suitable when we believe that there is substantial redundancy among the parameters in the sense that they are highly correlated. Thus, since it is likely that adjacent points on the volatility surface move together, PCA should be able to reduce the number of observed variables into a few principal components which are more easily interpreted.

Essentially, principle component analysis is an application of basic linear algebra that provides a representation for a high dimensional random vector with correlated components in terms of a factor model with fewer uncorrelated factors. Assume that we have n observations of m parameters, where the parameters are the differences between daily observations of the volatility surface. Write $\mathbf{f} = (f_1, ..., f_m)^T$ and $\Sigma_{\mathbf{f}} = \mathrm{Cov}(\mathbf{f})$. Recall that the symmetric and positive semidefinite matrix $\Sigma_{\mathbf{f}}$ may be expressed as the product $\Sigma_{\mathbf{f}} = ODO^T$, where D is a diagonal matrix with the (nonnegative) eigenvalues $\lambda_1, ..., \lambda_m$ of $\Sigma_{\mathbf{f}}$ as diagonal elements and O is an orthogonal matrix whose columns $\mathbf{o}_1, ..., \mathbf{o}_m$ are eigenvectors of $\Sigma_{\mathbf{f}}$, orthogonal and of length one. These columns are the principal components, which means that the number of principal components is equal to the number of observed variables. However, in most analyses, only the first few components account for meaningful amounts of variance, which is why the method is commonly used. The principal component analysis can either be performed by finding the matrices O and D, or by the Matlab command *princomp*, which return the columns of O as well as the diagonal elements of D in descending order.

Chapter 4

Calibration Results

In this section, we present the main results of the empirical study. We start out by presenting the estimated model parameters of the Heston model and discuss their validity. Secondly, we discuss the parameter variations over time. Thirdly, we present the results from the calibration of the Dupire model. In particular, we discuss the results from the principal component analysis, to illustrate how the local volatility surface varies over time.

4.1 The Heston Model

4.1.1 Parameter Estimates

The average parameter estimates from the 134 daily estimations are shown in Table 4.1. Firstly, we note that the choice of loss function has a significant impact on the parameter estimates. For obvious reasons, this constitutes a major problem when calibrating the Heston model, as there are no general guidelines for choosing the loss function. As mentioned earlier, Detlefssen and Härdle suggest using the IV MSE loss function, when we are fairly certain that the underlying model is correct. On the other hand, if there is uncertainty about the correctness of the model, the $ MSE loss function is preferable.

Table 4.1: Average parameter estimates

Loss function	κ	θ	σ	ρ	V_0
% MSE	2.2319	0.0515	0.4829	-0.8960	0.1515
$ MSE	3.8676	0.0829	0.3257	-0.9985	0.0790
IV MSE	4.4741	0.0686	0.3296	-0.9992	0.0891

Moving on to the validity of the parameters, several interesting characteristics can be observed. Firstly, the spot volatilities for the three loss functions all lie in the range of 28-39 %, which is slightly higher than for example Christoffersen, Heston and Jacobs (2009). These higher values can probably be explained by the financial crisis last year, leading to large fluctuations in the stock market. Furthermore, we note that the correlation between return and volatility is negative for all loss functions, which indicates that the Heston model is able to generate the observed smirk shape in volatility skew. It is worth noticing that the correlation coefficient is very close to -1 for all error functionals. This suggests that the volatility may be modeled as a deterministic function of the underlying asset price, and the assumption of stochastic volatility might in fact be unnecessary.

The estimated long-run mean of the stochastic variance process is also in accordance with earlier studies, with an average long-run mean volatility in the interval 22-29 % (note that the long-run mean volatility is defined as $\sqrt{\theta}$). The value of the average long-run mean volatility is also fairly constant, independent of the choice of loss function. Turning to the different estimates of the mean reversion parameter κ, the values vary between 2.2 and 4.5. These values coincide with the estimates obtained by Christoffersen, Heston and Jacobs (2009) as well as with Bakshi, Cao and Zen (1997).

All-in-all, the parameter estimates of the Heston model are in line with our expectations as well as the results of earlier empirical studies. Thus, we proceed to investigate the parameter variations over time to determine the robustness of the Heston model with respect to market conditions.

4.1.2 Parameter Variations over Time

Despite the fact that the choice of loss function seems to have a large impact on the parameter estimates, the parameter variations over time follow a similar pattern independent of the error functional. Therefore, while presenting the results for all three loss functions, we restrict our analysis mainly to the % MSE loss function. Looking at the plots presented in figure 4.1, several interesting characteristics are revealed. Firstly, we note that the observed time period may be divided into two different periods where the behavior of the parameter estimates differ significantly. In particular, this partition seem to coincide with the partition of the market into one part of stable movements (first 60 days) and a second part of larger fluctuations in the underlying asset (last 70 days). Beginning with the first period of time, we note that when the price of the underlying asset is fairly stable, the parameter estimates are rather stable as well. The 20-day period of higher values of θ and ρ might be caused by the calibration process. Overall, this indicates that at times when there are only small fluctuations in the market, the stochastic part of the Heston model (i.e. the two Wiener processes) are sufficient in explaining these movements, while the parameters remain almost constant. On the other hand, we note that in the second period when the market becomes more volatile, the parameter estimates fluctuate heavily. Thus, there are indications that the stochastic part is insufficient in explaining large variations

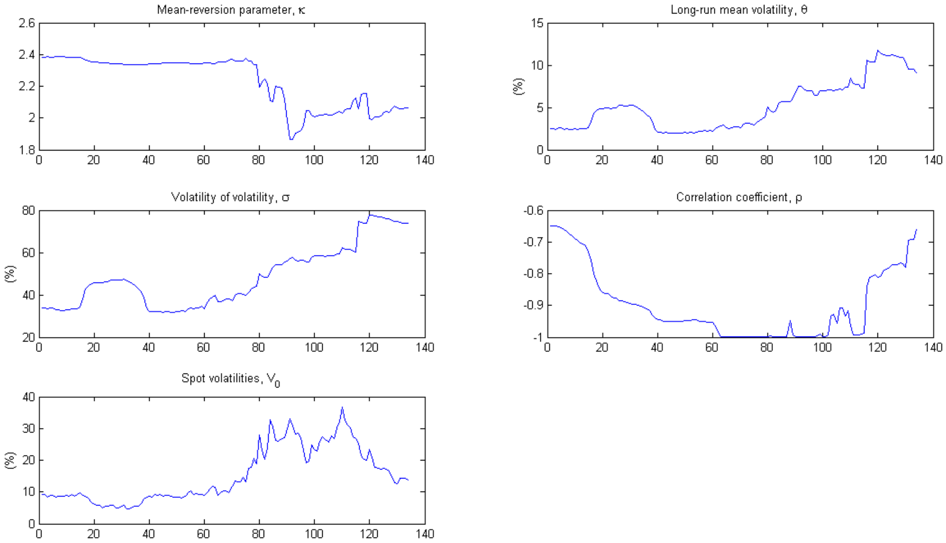

Figure 4.1: Variations in parameter estimates using the % MSE loss function.

in the market.

To develop a more thorough understanding of the parameter dynamics we take a closer look at the parameter variations. Firstly, we observe that when the market becomes more volatile, the value of the mean reversion parameter κ decreases. Thus, the volatility process becomes less mean reverting, allowing the volatility to deviate more from its long-run mean. Looking at the graph of the spot volatilities, we see that this is exactly what occurs. A high degree of mean reversion (i.e. higher value of κ) means that the volatility is rapidly mean reverting, leading to fast oscillations around its long-run mean value. This is captured by fairly low and stable spot volatilities in the first period, while the second time period presents both higher values and larger fluctuations in the spot volatilities.

Moving on to the long-run mean volatility, we note that as the volatility of the market increases, so does the value of θ. Perhaps this is not surprising, but since θ is a long-run mean parameter we would expect a robust model to allow fluctuations in the underlying asset without changing the value of θ. In addition, it seems plausible to assume that changes in the long-run mean volatility occur slowly over time, and not instantaneously as market conditions change. However, this assumption is contradicted by the fact that the behavior of θ follow the behavior of the parameter σ, which is the volatility of the volatility. The fact that the values of σ tend to increase when the underlying asset price fluctuates heavily is naturally in accordance with prior expectations.

Finally, we turn to the value of the correlation between return and volatility. Having noted earlier that the value of ρ affects the skewness distribution, we would expect the correlation coefficient to become more negative when markets are volatile. Looking at the graph of ρ we note that the actual outcome is in line with our expectations. When the underlying asset price fluctuate the value of ρ decreases, thus creating a fat left-tailed distribution. All in all, the Heston model is consistent in the sense that the parameter variations can be readily explained by the current market conditions.

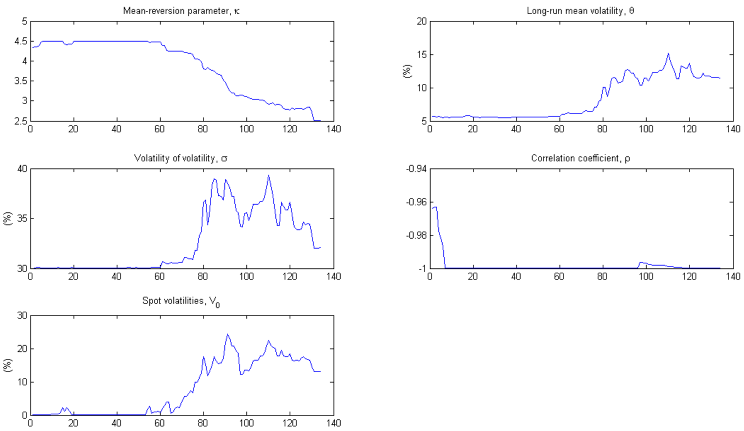

Figure 4.2: Variations in parameter estimates using the \$ MSE loss function.

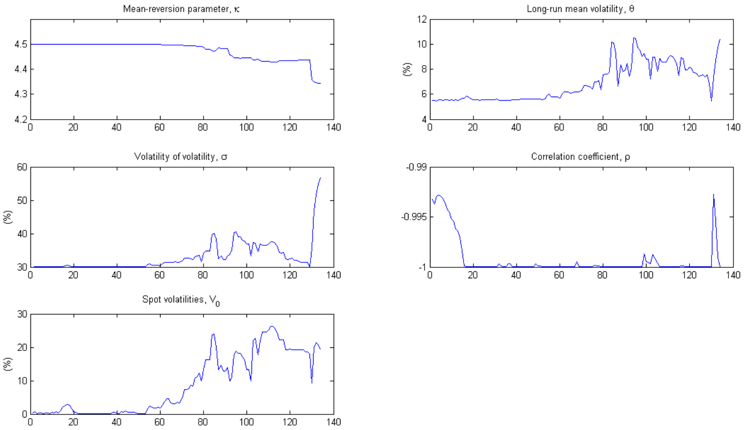

Figure 4.3: Variations in parameter estimates using the IV MSE loss function.

If we briefly consider the variations of the parameter estimates from the \$ MSE function and the IV MSE function (see figures 4.2 and 4.3) we note that the patterns are very similar. This indicates that the time inconsistency present in the Heston model is not due to the calibration method, but rather an inherent feature of the model. In the next section, we look closer at the robustness of the Dupire model, and in particular the results obtained from the principal component analysis.

4.2 The Dupire Model

Unlike the Heston model, where there are a large number of papers on the calibration process, the calibration of the Dupire model required several steps of trial and error. Since the implicit method is known to give more stable solutions, the explicit method was never implemented in practice. Furthermore, the $ MSE loss function was used during the calibrations. The appropriate values of the regularization parameters α_K and α_T were also determined using trial and error. It turns out that for the given market data $\alpha_K = 10^7$ and $\alpha_T = 10$ give good prices and sufficiently smooth volatility surfaces. The large difference between the two parameters is simply due to the fact that the value of the second derivative with respect to K is in general much smaller than the corresponding second derivative with respect to T. The effects of adding the regularization terms are illustrated in figures 4.4 and 4.5 below.

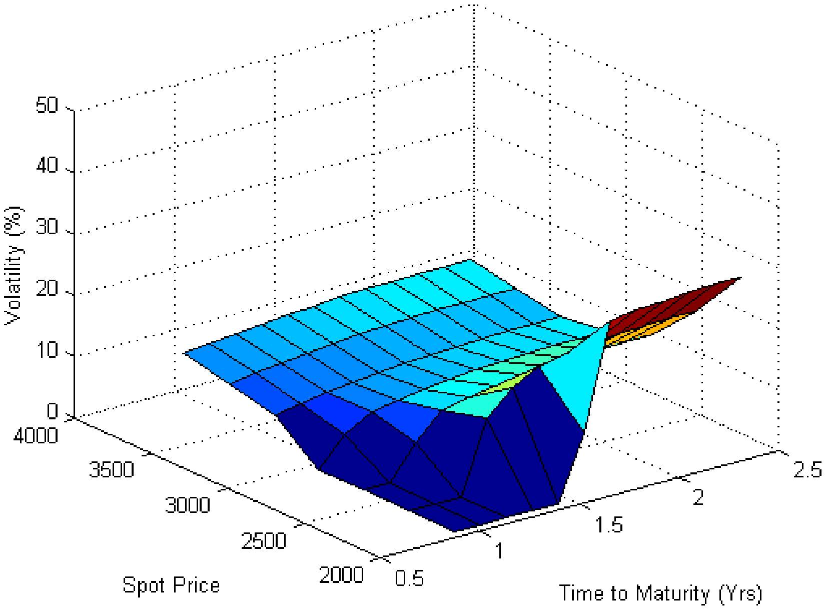

Figure 4.4: Local volatility function on June 23^{rd} when no regularization term is added.

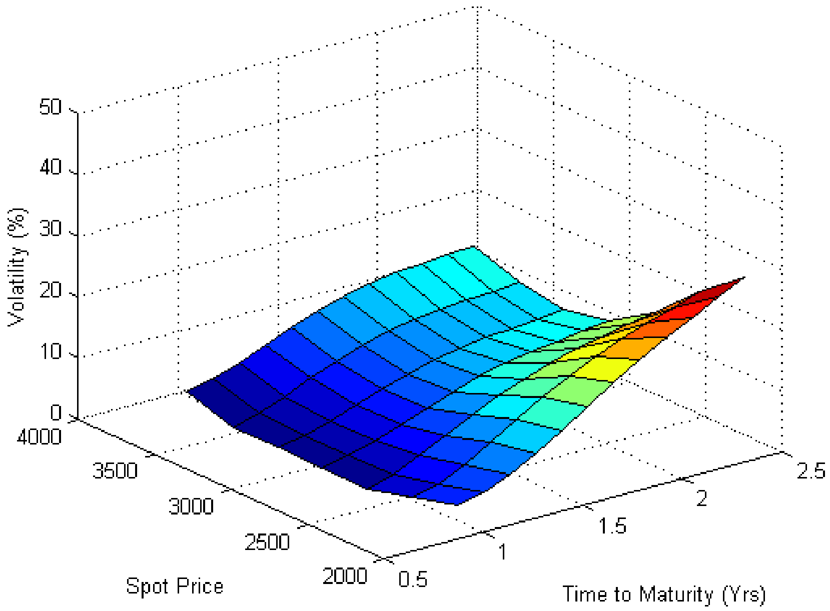

Figure 4.5: Local volatility function on June 23^{rd} when a regularization term is added.

Care should be taken not to confuse the local volatilies with implied volatilities. However, the two are closely connected. Gatheral (2006) shows that there is a quite intuitive picture for the meaning of Black-Scholes implied variance of a European option with a given strike and expiration: It is approximately the integral from today to expiration of local variances along the most probable path for the stock price conditional on the stock price at expiration being the strike price of the option. This mental picture should be kept in mind as we analyse the dynamics of the local volatility function.

Moving on to the variation of the local volatility function over time, we note that in accordance with the Heston model, there are two distinct time periods where the properties of the model vary significantly. During the first period (first 60 days), when the price of the underlying asset is quite stable, the shape of the volatility surface is fairly constant. We also note that the model is able to capture the volatility smirk that has been readily observed in the market. The volatility function is also fairly stable with respect to the time to maturity, which is in line with prior expectations.

In the second period, when the fluctuations in the market increase in amplitude, the shape of the volatility surface is heavily distorted (see figure 4.7), with out-of-the-money volatilities approaching zero. Essentially, this is equivalent to saying that the market's expectations of a rise in the price of underlying are very low. Consequently, the prices of out-of-the-money

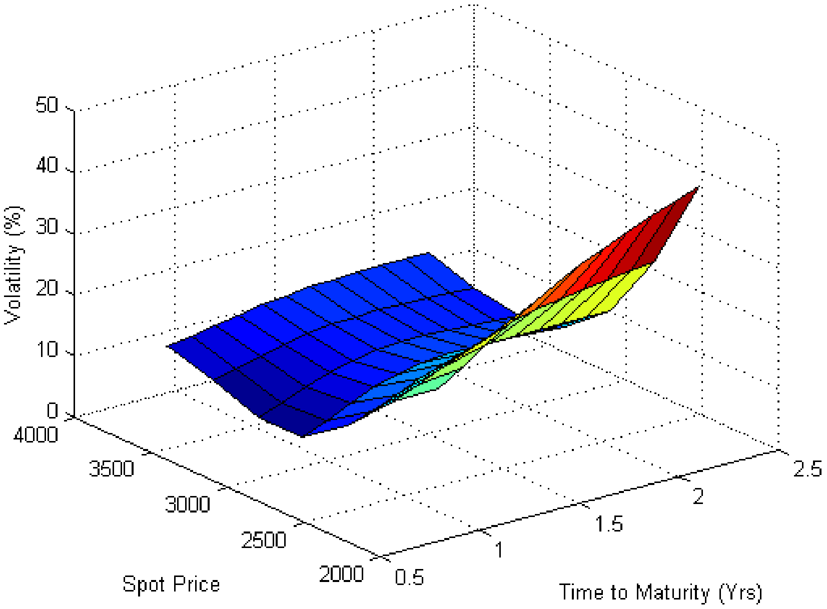

Figure 4.6: Local volatility function on September 11^{th}

calls will drop as well as the corresponding volatilities. Moreover, the fact that the shape of the local volatility function varies considerably over time, provides a first indicator of time inconsistency in the Dupire model.

A more thorough understanding of the underlying dynamics of the volatility surface is provided by the principal component analysis. The Matlab function *princomp* was used to calculate the principal components of the surface variations, and the cumulative variances of the first six components are shown in table 4.2 below. Consequently, we establish that the three first principal components account for almost 90 % of the total variance in the local volatility function. A closer look at these principal components (see Appendix A) reveals that the variance of the volatility surface to a large

Table 4.2: Cumulative variances of the first six principal components.

Principal component	Cumulative variance
1	0.44629
2	0.8002
3	0.88821
4	0.94017
5	0.96703
6	0.98439

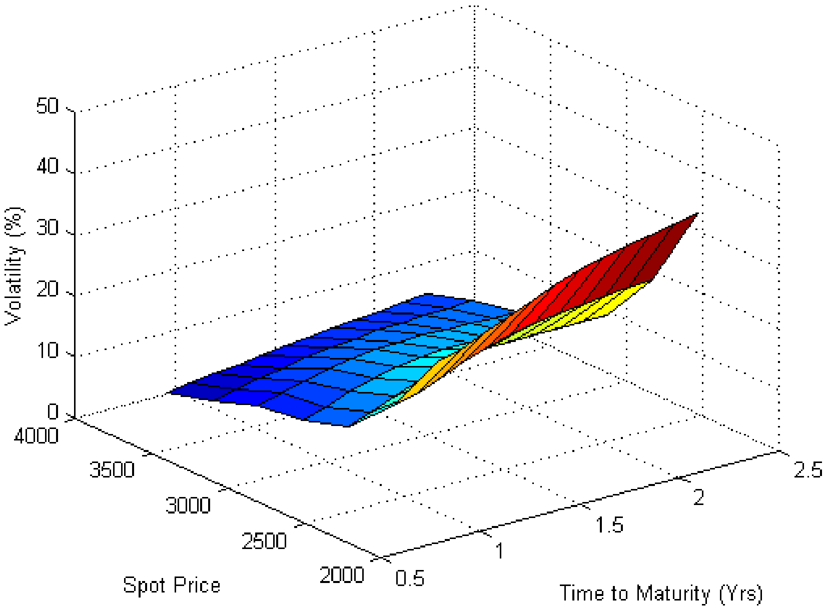

Figure 4.7: Local volatility function on September 16^{th}

extent can be explained by three distinct movements. The first principal component shows that over 40 % of the variance is due to vertical shifts of the volatility surface. Moreover, the local volatilities at low spot prices tend to shift slightly more than volatilities for high spot prices. This suggests than volatilities of out-of-the-money options are more stable than the corresponding in-the-money options. This phenomenon can partly be explained by the fact that, in the second time period, volatilities of out-of-the-money options are stable and close to zero. In addition, short maturity volatilities seem to shift slightly more than long maturity volatilities.

The second principal component indicates that another 36 % of the variance can be explained by the fact that when volatilities of in-the-money increase, the corresponding volatilities of out-of-the-money decrease, and vice versa. This movement of the volatility surface affects the volatility smirk, making it more or less prominent. The phenomenon is readily seen during the second part of the time period, when the price of the underlying asset decreases.

Finally, the third component shows that almost 10 % of the variability is due to the fact that when volatilities of in-the-money and out-of-the-money options increase, the corresponding volatilities of at-the-money options decrease, and vice versa. Basically, this means that when the market expects large fluctuations in the underlying asset price, volatilities of in-the-money and out-of-the-money options will increase as the corresponding option prices go up. In a volatile market, the demand for at-the-money options is likely to decrease, thus leading to a reduction of at-the-money volatilities.

So, in accordance with the results obtained from the Heston model, the Dupire model is robust in the sense that the observed surface variations can be readily explained by the underlying asset and the current market conditions. On the other hand, the model shows a high degree of time inconsistency, as the shape of the local volatility function tends to fluctuate heavily when the market is volatile. Moreover, the principal component analysis turns out to be an extremely useful tool in reducing the noise in parameter variations and explaining, in a self-consistent way, the dynamics of the Dupire model.

Chapter 5

Conclusions

This paper investigates the time consistency of two structural option pricing models, in terms of variations in daily parameter estimates. Our results show that the parameter variations can be readily explained by the current market conditions, while the fact that they vary indicate an inherent time inconsistency of the models. This is interesting for several reasons.

The models investigated are constructed based on the assumption that the long-term evolution of the underlying spot price and its volatility can be described using suitable diffusion processes. At some time t we calibrate the models to market data, expecting to find parameters that are consistent with the market's expectations of the future evolution of the underlying asset price, in the sense that they are able to recover option prices as accurately as possible. In a time consistent model, these parameter estimates completely specify the set of possible structures for any $t > 0$. Since the parameter estimates vary over time, we conclude that the main assumptions of true underlying diffusion processes are inaccurate, partly agreeing with earlier research conducted by for example Christoffersen and Jacobs (2004).

In summary, this thesis provides valuable insights regarding the dynamics of current option pricing models. At the same time, the inconsistencies arising from the varying parameter estimates, as well as the difficulties in calibrating the models, shows that much of the option pricing theory is yet to be discovered, both regarding model specification and implementation.

Bibliography

[1] Bakshi, G., Cao, Charles., and Chen, Z. 1997. Empirical Performance of Alternative Option Pricing Models. *The Journal of Finance* 52, 2003 - 2049.

[2] Björk, T., *Arbitrate Theory in Continuous Time.* Oxford University Press, New York, 2 edition, 2004.

[3] Black, F., and Scholes, M. 1973. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* 81, 653 - 659.

[4] Brecher, D. 2006. Pushing the Limits of Local Volatility in Option Pricing. *Wilmott Magazine*, pp. 6 - 15.

[5] Carr, P., and Madan, D. 1999. Option valuation using the Fast Fourier Transform.

[6] Christoffersen, P., Heston, S., and Jacobs, K. 2009. The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well.

[7] Christoffersen, P., Jacobs, K., *The Importance of the Loss Function in Option Valuation*, Journal of Financial Economics 2004 Volume 72, pp. 291 - 318

[8] Cont, R., and Hamida, S. 2005. Recovering volatility from option prices by evolutionary algorithm. *Journal of Computational Finance* 8.

[9] Cox, John C., Jonathan E. Ingersoll, and Steven A. Ross. 1985. A theory of the term structure of interest rates. *Econometrica* 53, 385 - 407.

[10] Cox, John C., and Ross, Stephen A. 1976, *The Valuation of Options for Alternative Stochastic Processes*, Journal of Financial Economics 8, 43 - 76.

[11] Derman, E., and Kani, I. 1994. Riding on a Smile. *Risk* 7, 32 - 39.

[12] Detlefsen, K., Härdle, W., Discussion Paper, 2006

[13] Dupire, B. 1993. Pricing and Hedging with a Smile.

[14] Dupire, B. 1994. Pricing with a Smile. *Risk* 7, 18 - 20.

[15] Elder, John. 2002. Hedging for financial derivatives. *University of Oxford.* Ph.D. Thesis.

[16] Feller, W. 1951. Two singular diffusion problems. *The Annals of Mathematics* 54, 173-182.

[17] Gatheral, J., *The Volatility Surface - A Practitioners Guide*, John Wiley & Sons, Inc.

[18] Hamida, C., Cont, R.,*Recovering Volatility from Option Prices by Evolutionary Optimization*

[19] Heston S. 1993. A closed-form solution for options with stochastic volatility, with application to bond and currency options. *Review of Financial Studies* 6, 327 - 343.

[20] Mikhailov, S., and Nögel, U. 2003. Heston's stochastic volatility model, calibration and some extensions. *Wilmott Magazine*, pp. 74 - 79.

[21] Schoutens, Wim., Simons, Erwin., and Tistaert, Jurgen. 2003. A Perfect Calibration! Now What?

Appendix A

Results

A.1 Principal Component Analyis

```
PrincipalComponents =

   -0.2212    0.1028    0.2083
   -0.1891    0.1211   -0.1601
   -0.1350    0.0688   -0.3604
   -0.1111   -0.0275   -0.2220
   -0.1293   -0.0800    0.0995
   -0.2386    0.1266    0.1500
   -0.1862    0.1172   -0.1644
   -0.1355    0.0526   -0.3235
   -0.1181   -0.0438   -0.1958
   -0.1349   -0.0949    0.1011
   -0.2416    0.1419    0.1040
   -0.1727    0.1059   -0.1495
   -0.1284    0.0292   -0.2678
   -0.1217   -0.0640   -0.1566
   -0.1382   -0.1111    0.1113
   -0.2335    0.1494    0.0763
   -0.1514    0.0893   -0.1206
   -0.1148    0.0021   -0.2073
   -0.1207   -0.0866   -0.1166
   -0.1388   -0.1284    0.1222
   -0.2182    0.1508    0.0668
   -0.1263    0.0705   -0.0848
   -0.0976   -0.0253   -0.1525
   -0.1159   -0.1097   -0.0838
   -0.1366   -0.1465    0.1290
   -0.1994    0.1482    0.0719
   -0.1015    0.0524   -0.0492
   -0.0799   -0.0500   -0.1095
   -0.1084   -0.1315   -0.0614
```

```
-0.1323   -0.1648    0.1299
-0.1803    0.1436    0.0864
-0.0798    0.0368   -0.0183
-0.0641   -0.0702   -0.0798
-0.0998   -0.1509   -0.0492
-0.1268   -0.1830    0.1250
-0.1629    0.1384    0.1050
-0.0629    0.0247    0.0058
-0.0514   -0.0857   -0.0614
-0.0909   -0.1672   -0.0448
-0.1208   -0.2007    0.1155
-0.1482    0.1334    0.1243
-0.0509    0.0158    0.0237
-0.0420   -0.0969   -0.0509
-0.0823   -0.1808   -0.0449
-0.1149   -0.2178    0.1033
-0.1361    0.1288    0.1422
-0.0428    0.0095    0.0370
-0.0351   -0.1053   -0.0449
-0.0741   -0.1925   -0.0472
-0.1095   -0.2343    0.0897
-0.1257    0.1245    0.1587
-0.0372    0.0046    0.0480
-0.0295   -0.1121   -0.0408
-0.0662   -0.2033   -0.0500
-0.1044   -0.2504    0.0757
-0.1161    0.1204    0.1744
-0.0325    0.0002    0.0582
-0.0244   -0.1184   -0.0371
-0.0582   -0.2137   -0.0526
-0.0995   -0.2663    0.0617
```

Appendix B

Matlab code

B.1 Calibrating the Heston model using Differential Evolution

```
%%HestonDECalibration uses a Differential Evolution Algorithm to fit the
%%parameters of the Heston model to market option prices.
%%The optimization scheme can be downloaded from
%%http://www.mathworks.com/matlabcentral/fileexchange/18593

clear, close all
clc

global OptionData; global NoOfOptions; global PriceDifference;
global ImplBSVol; global InitialParameters;

load RawOptionData.m; %% r - D, T, S0, K, C, r, D

%CALCULATE THE NUMBER OF DATES AND THE NUMBER OF OPTIONS FOR EACH DATE
[B,I,J] = unique(RawOptionData(:,3),'first');
Index = sort(I); Index(length(Index) + 1) = length(RawOptionData(:,3));
for i=1:length(Index)-1
    NoOptions(i) = Index(i+1)-Index(i);
end
NoOptions(length(NoOptions)) = NoOptions(length(NoOptions))+1;

NoOfDates = length(NoOptions);
NUsed = 1;

%SET INITIAL PARAMETERS: [Kappa;Theta;Sigma;Rho;V0]
InitialParameters = [2.3846;0.02522;0.33714;-0.64819;0.09163];

Results = [];
for j=1:NoOfDates
```

```
    NoOfOptions = NoOptions(j);
        OptionData = RawOptionData([NUsed:NUsed+NoOfOptions-1],:);

%%IF WE USE THE IV MSE LOSS FUNCTION, CALCULATE THE IMPLIED
%%BLACK-SCHOLES VEGA: OTHERWISE, THIS SECTION MAY BE
%%IGNORED
%for k=1:NoOfOptions
%y = OptionData(k,:);
%ImplBSVol(k) = fzero(@(x) bsvolatility(x,y),0.2);
%end
%for l=1:NoOfOptions
%OptionData(l,8) = blsvega(OptionData(l,3),OptionData(l,4),...,
%OptionData(l,6),OptionData(l,2),ImplBSVol(l),OptionData(l,7));
%end

%SPECIFY PARAMETERS FOR THE DIFFERENTIAL EVOLUTION OPTIMIZATION
optimInfo.title = 'Heston Differential Evolution Calibration';
objFctHandle = @HestonCostFunc;
paramDefCell = {'',[0.5 7;0.001 1;0.001 1;-1 1;0.001 0.5],...,
[1e-5*ones(5,1)],InitialParameters};
objFctSettings = {};
objFctParams = [];
DEParams = getdefaultparams;
DEParams.NP = 100;
DEParams.feedSlaveProc = 0;
DEParams.validChkHandle = @HestonConstraint; %2*Kappa*Theta>=0
DEParams.maxiter = 1e5;
DEParams.maxtime = 600; %Maximum optimization time (in secs)
DEParams.maxclock = [];
DEParams.refreshiter = 1;
DEParams.refreshtime = 10;
DEParams.refreshtime2 = 600;
DEParams.refreshtime3 = 1200;
emailParams = [];
rand('state',1);

%bestmem = PARAMETER ESTIMATES, bestval = LOSS FUNCTION VALUE
[bestmem,bestval] = differentialevolution(DEParams,paramDefCell,...,
objFctHandle,objFctSettings,objFctParams,emailParams,optimInfo);

NUsed = NUsed + NoOfOptions;
InitialParameters = bestmem;
Results = [Results;bestmem' bestval];
end
```

```
function val = HestonCostFunc(x, noPause)

global OptionData;
global NoOfOptions;
global PriceDifference;
global ImplBSVol;
global InitialParameters;

%% CHOOSE THE LOSS FUNCTION TO BE USED AND THE REGULARIZATION PARAMETER
%% IV-MSE LOSS FUNCTION
% for i = 1:NoOfOptions
%     PriceDifference(i) = (OptionData(i,5) - ...,
%         HestonCallQuad(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
%         OptionData(i,2),OptionData(i,3),OptionData(i,4)))/...,
%        ((blsvega(OptionData(i,3),OptionData(i,4),OptionData(i,6),...,
%         OptionData(i,2),ImplBSVol(i),OptionData(i,7))));
% end
% RegularizationParameter = 1;
% val = sqrt(1/NoOfOptions^2*sum(PriceDifference.^2)) + ...,
% RegularizationParameter*sum((InitialParameters-x).^2);


%% $-MSE LOSS FUNCTION
% for i = 1:NoOfOptions
%     PriceDifference(i) = (OptionData(i,5) - ...,
%         HestonCallQuad(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
%         OptionData(i,2),OptionData(i,3),OptionData(i,4)));
% end
% RegularizationParameter = 1;
% val = sqrt(1/NoOfOptions^2*sum(PriceDifference.^2)) + ...,
% RegularizationParameter*sum((InitialParameters-x).^2);
%% %-MSE LOSS FUNCTION
for i = 1:NoOfOptions
    PriceDifference(i) = (OptionData(i,5) - ...,
        HestonCallPrice(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
        OptionData(i,2),OptionData(i,3),OptionData(i,4)))/OptionData(i,5);
end
RegularizationParameter = 1;
val = sqrt(1/NoOfOptions^2*sum(PriceDifference.^2)) + ...,
RegularizationParameter*sum((InitialParameters-x).^2);



function call = HestonCallPrice(kappa,theta,sigma,rho,v0,r,T,s0,K)

warning off;
call = s0*HestonProbabilities(kappa,theta,sigma,rho,v0,r,T,s0,K,1)- ...,
K*exp(-r*T)*HestonProbabilities(kappa,theta,sigma,rho,v0,r,T,s0,K,2);
```

```
function ret = HestonProbabilities(kappa,theta,sigma,rho,v0,r,T,s0,K,type)
ret = 0.5 + 1/pi*quadl(@HestonIntegrand,0,100,[],[],kappa,theta,sigma,...,
rho,v0,r,T,s0,K,type);
```

```
function ret = HestonIntegrand(phi,kappa,theta,sigma,rho,v0,r,T,s0,K,type)
 ret = real(exp(-i*phi*log(K)).*HestonCharFunc(phi,kappa,theta,sigma,...,
rho,v0,r,T,s0,type)./(i*phi));
```

```
function f = HestonCharFunc(phi,kappa,theta,sigma,rho,v0,r,T,s0,type);
if type == 1
    u = 0.5;
    b = kappa - rho*sigma;
else
    u = -0.5;
    b = kappa;
end

a = kappa*theta;
x = log(s0);
d = sqrt((rho*sigma*phi.*i-b).^2 - sigma^2*(2*u*phi.*i-phi.^2));
g = (b - rho*sigma*phi*i + d)./(b - rho*sigma*phi*i - d);
C = r*phi.*i*T + a/sigma^2.*((b - rho*sigma*phi*i + d)*T - ...,
2*log((1-g.*exp(d*T))./(1-g)));
D = (b - rho*sigma*phi*i + d)./sigma^2.*((1-exp(d*T))./(1-g.*exp(d*T)));
f = exp(C + D*v0 + i*phi*x);
```

B.2 Calibrating the Heston model using *lsqnonlin*

```
clear, close all
clc

global NoOfOptions; global OptionData;
global InitialParameters;

load RawOptionData.m; %% r - D, T, S0, K, C, r, D

%CALCULATE THE NUMBER OF DATES AND THE NUMBER OF OPTIONS FOR EACH DATE
[B,I,J] = unique(RawOptionData(:,3),'first');
Index = sort(I); Index(length(Index) + 1) = length(RawOptionData(:,3));
```

```
for i=1:length(Index)-1
    NoOptions(i) = Index(i+1)-Index(i);
end
NoOptions(length(NoOptions)) = NoOptions(length(NoOptions))+1;

NoOfDates = length(NoOptions);
NUsed = 1;

%SET INITIAL PARAMETERS: [Kappa;Theta;Sigma;Rho;V0]
InitialParameters = [2.3846;0.02522;0.33714;-0.64819;0.09163];

Results = [];
for j=1:NoOfDates
    NoOfOptions = NoOptions(j);
    OptionData = RawOptionData([NUsed:NUsed+NoOfOptions-1],:);

%%IF WE USE THE IV MSE LOSS FUNCTION, CALCULATE THE IMPLIED
%%BLACK-SCHOLES VEGA: OTHERWISE, THIS SECTION MAY BE
%%IGNORED
%for k=1:NoOfOptions
%y = OptionData(k,:);
%ImplBSVol(k) = fzero(@(x) bsvolatility(x,y),0.2);
%end
%for l=1:NoOfOptions
%OptionData(l,8) = blsvega(OptionData(l,3),OptionData(l,4),...,
%OptionData(l,6),OptionData(l,2),ImplBSVol(l),OptionData(l,7));
%end

bestmem = lsqnonlin(@(x) HestonDifferences(x),...,
InitialParameters,[0.5 7;0.001 1;0.001 1;-1 1;0.001 0.5]);
Results = [Results;bestmem'];
NUsed = NUsed + NoOfOptions;
InitialParameters = bestmem;
end



function ret = HestonDifferences(x)
global NoOfOptions; global OptionData;
global InitialParameters;
%% CHOOSE THE LOSS FUNCTION TO BE USED AND THE REGULARIZATION PARAMETER
%% IV-MSE LOSS FUNCTION
% for i = 1:NoOfOptions
%     PriceDifference(i) = (OptionData(i,5) - ...,
%         HestonCallQuad(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
%         OptionData(i,2),OptionData(i,3),OptionData(i,4)))/...,
%        ((blsvega(OptionData(i,3),OptionData(i,4),OptionData(i,6),...,
%         OptionData(i,2),ImplBSVol(i),OptionData(i,7))));
% end
% RegularizationParameter = 1;
% for j=1:5
%     PriceDifference(NoOfOptions+j) = sqrt(RegularizationParameter)*...
%(InitialParameters(j)-x(j));
% end
% ret = PriceDifference;



%% $-MSE LOSS FUNCTION
% for i = 1:NoOfOptions
%     PriceDifference(i) = (OptionData(i,5) - ...,
%         HestonCallQuad(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
%         OptionData(i,2),OptionData(i,3),OptionData(i,4)));
% end
% RegularizationParameter = 1;
% for j=1:5
%     PriceDifference(NoOfOptions+j) = sqrt(RegularizationParameter)*...
%(InitialParameters(j)-x(j));
% end
% ret = PriceDifference;
%% %-MSE LOSS FUNCTION
for i = 1:NoOfOptions
    PriceDifference(i) = (OptionData(i,5) - ...,
        HestonCallPrice(x(1),x(2),x(3),x(4),x(5),OptionData(i,1),...,
        OptionData(i,2),OptionData(i,3),OptionData(i,4)))/OptionData(i,5);
end
RegularizationParameter = 1;
for j=1:5
    PriceDifference(NoOfOptions+j) = sqrt(RegularizationParameter)*...
(InitialParameters(j)-x(j));
end
ret = PriceDifference;
```

B.3 Calibrating the Dupire model using *lsqnonlin*

```
clear, close all
clc

global S0; global K; global NPartition; global dT; global dK;
global u; global AlphaK; global AlphaT; global CallObservations;

load RawOptionData.m;
[B,I,J] = unique(RawOptionData(:,3),'first');
Index = sort(I); Index(length(Index) + 1) = length(RawOptionData(:,3));
for i = 1:length(Index)-1
```

```
    NoOfOptions(i) = Index(i+1)-Index(i);
end
NoOfOptions(length(NoOfOptions)) = NoOfOptions(length(NoOfOptions)) + 1;
NoOfDates = length(NoOfOptions); NUsed = 1;

NPartition = 20;
InitialVol = 0.2*ones(NPartition-1,NPartition-2); %INITIAL PARAMETERS
Results = []; TVec = []; KVec = [];

AlphaK = 10^7; AlphaT = 10;

for i=1:NoOfDates
    OptionData = RawOptionData([NUsed:NUsed+NoOfOptions(i)-1],:);
    S0 = OptionData(1,3);

    Tmin = min(OptionData(:,2)); Tmax = max(OptionData(:,2));
    dT = Tmax/(NPartition-1); T = 0:dT:Tmax;

    Kmin = 0; Kmax = 2*S0;   %Note that the value of Kmax may vary between
    dK = Kmax/(NPartition-1);%different dates. Choose Kmax sufficiently
    K = 0:dK:Kmax;           %large, so that we may set the value of
                             %options with K = Kmax to 0.
    t1 = OptionData(:,2); k1 = OptionData(:,4); c1 = OptionData(:,5);
    TUnique = unique(t1); KUnique = unique(k1);
    k2 = zeros(length(TUnique),1); k3 = Kmax*ones(length(TUnique),1);
    tk_initial = [t1 k1 c1;
                  TUnique k2 S0*ones(length(TUnique),1);
                  TUnique k3 zeros(length(TUnique),1)];
    tk = tk_initial(:,[1:2]); c = tk_initial(:,3);
    calls = tpaps(tk',c');

    for j=1:length(T)
        for k=1:length(K)
            CallObservations(j,k) = max(fnval(calls,[T(j);K(k)]),0);
        end
    end

    r_D = [OptionData(:,2) OptionData(:,1)];
    rates = sortrows(unique(r_D,'rows'));
    u = interp1(rates(:,1),rates(:,2),T,'pchip');

    LVFunc = lsqnonlin(@(v) ImplicitDupire(v),InitialVol,zeros(NPartition-1,...
    ,NPartition-2),ones(NPartition-1,NPartition-2));
    Results = [Results;LVFunc]; TVec = [TVec;T]; KVec = [KVec;K];
    InitialVol = LVFunc;
    NUsed = NUsed + NoOfOptions(i);
end
```

```
function ret = ImplicitDupire(Vol)

global S0; global K; global NPartition; global dT; global dK;
global u; global AlphaK; global AlphaT; global CallObservations;

C(1,1) = S0; C(1,NPartition) = 0;
for i=2:NPartition-1
    C(1,i) = max(S0-K(i),0);
end
for t=2:NPartition
    A(1,1)=1; A(NPartition,NPartition)=1;
    for k=2:NPartition-1
        A(k,k-1) = -0.5*Vol(t-1,k-1)^2*K(k)^2*dT/dK^2;
        A(k,k) = 1 + Vol(t-1,k-1)^2*K(k)^2*dT/dK^2 - u(t)*K(k)*dT/dK;
        A(k,k+1) = u(t)*K(k)*dT/dK - 0.5*Vol(t-1,k-1)^2*K(k)^2*dT/dK^2;
    end
    B = sparse(A);
    b = [S0;C(t-1,[2:NPartition-1])';0];
    c = B\b;
    C(t,:) = c';
end
PriceDifference = CallObservations([2:NPartition],[2:NPartition-1])-...
C([2:NPartition],[2:NPartition-1]);
for i=1:NPartition-1
    for j=1:NPartition-4
        VolKK(i,j) = (Vol(i,j+2)-2*Vol(i,j+1)+Vol(i,j))/dK^2;
    end
end
for i=1:NPartition-2
    for j=1:NPartition-3
        VolTT(i,j) = (Vol(j+2,i)-2*Vol(j+1,i)+Vol(j,i))/dT^2;
    end
    VolTT(NPartition-1,:) = 0;
end
ret = [PriceDifference AlphaK*VolKK AlphaT*VolTT];
```